Machine Learning Anomaly Detection Applications to Compact Muon Solenoid Data Quality Monitoring

Adrian Alan Pol

université
PARIS-SACLAY

Machine Learning Anomaly Detection Applications to Compact Muon Solenoid Data Quality Monitoring

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 : sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique
Unité de recherche : Université Paris-Saclay, CNRS, Laboratoire de recherche en informatique, 91405, Orsay, France
Référent : Faculté des sciences d'Orsay

Thèse présentée et soutenue à Orsay, le 8 Juin 2020, par

Adrian Alan POL

Composition du Jury:

Anne VILNAT Professeure, Université Paris-Saclay	Présidente
Patrick GALLINARI Professeur, LIP6, CNRS, Sorbonne Université	Rapporteur & Examinateur
Danilo JIMENEZ REZENDE Docteur (HDR), Google Deep Mind	Rapporteur & Examinateur
Bertrand BRAUNSCHWEIG Docteur, Directeur de recherche, INRIA	Examinateur
Maurizio PIERINI Docteur, CERN	Examinateur
Slava VOLOSHYNOVSKIY Professeur, Université de Gèneve	Examinateur
Cécile GERMAIN-RENAUD Professeure, LRI, CNRS, Université Paris-Saclay	Directrice de thèse
Gianluca CERMINARA Docteur, CERN	Co-Encadrant

Thèse de doctorat

NNT : 2020UPASS083

dla rodziców

Résumé

Les expériences de physique des hautes énergies (HEP) impliquent des appareils de détection volumineux et complexes, qui doivent fonctionner avec une haute disponibilité pour profiter au mieux du temps de faisceau coûteux fourni par les accélérateurs de particules modernes. Pour traquer des phénomènes statistiquement rares, l'analyse des données de la physique des hautes énergies repose ultimement sur la connaissance approfondie de l'équipement expérimental. Cette méthode exige une surveillance constante et fiable pour détecter d'éventuels dysfonctionnements. Pour cette raison, toutes les collaborations HEP construisent des infrastructures et des procédures complexes de surveillance de la qualité des données, visant à identifier rapidement les anomalies résultant de problèmes matériels ou de traitement de données. L'expérience CMS au Large Hadron Collider (LHC) du CERN fonde ses procédures d'évaluation de la qualité sur la surveillance continue effectuée par des experts des détecteurs, qui comparent un grand nombre de tests statistiques avec ceux dérivés de références de bonne qualité. Cette méthode s'est avérée efficace pour identifier les problèmes et les anomalies pendant plus d'une décennie, mais elle atteint rapidement ses limites. La procédure repose sur la disponibilité d'un grand nombre de spécialistes de terrain hautement qualifiés. L'effort pour anticiper tous les types de pannes possibles avec des systèmes de règles basés sur des experts et un profilage statistique à l'aide d'histogrammes pose des problèmes de passage à l'échelle et plus généralement d'adaptabilité.

Cette thèse propose une approche possible de l'automatisation de la détection d'anomalies dans les expériences de physique des hautes énergies, en tenant compte des exigences et des contraintes spécifiques au domaine. Les problèmes de surveillance de la physique des hautes énergies sont généralement difficiles pour les méthodes statistiques conventionnelles en raison de leur nature intrinsèquement multidimensionnelle. L'apprentissage automatique et le sous-champ de détection des anomalies doivent être utilisés pour automatiser le processus de décision.

étant donné que le nombre de quantités surveillées et les modes de défaillance correspondants sont vastes et qu'aucune solution générale n'est possible, le présent travail se concentre sur certains des modèles de surveillance les plus fréquents dans l'expérience CMS Data Quality Monitoring.

Les réseaux de neurones convolutionnels sont utilisés pour analyser les images. Pour la surveillance en ligne des composants du sous-détecteur, nécessaire pour identifier les composants problématiques du détecteur avec une faible latence, un classifieur capable de détecter les comportements anormaux connus est proposé, ainsi que des méthodes pour étendre la couverture de surveillance actuelle en détectant de nouveaux modes de défaillance.

Les résultats montrent une efficacité sans précédent sur les modes de défaillance actuellement suivis. Les travaux ont couvert des aspects liés aux stratégies de mise à jour des modèles et à l'interprétation des résultats, qui sont d'une importance capitale dans un système qui devra être exploité pendant des années par des experts de terrain disposant d'une expertise limitée en apprentissage automatique.

Les autoencodeurs sont utilisés pour le contrôle de la qualité des données de collision du LHC représentées sous forme de distributions multidimensionnelles. Leur applicabilité a été démontrée la surveillance hors ligne pour l'identification de comportements problématiques nouveaux et émergents sur un grand nombre d'observables avec une granularité temporelle fine et des statistiques potentiellement faibles.

Des autoencodeurs variationnels conditionnels sont proposés pour la détection d'anomalies sur des données structurées hiérarchiquement. L'étude propose une nouvelle méthode AD-CVAE. Elle aborde un problème encore largement ouvert en apprentissage automatique: l'application des progrès rapides de la recherche sur les autoencodeurs variationnels à la détection d'anomalie, dans les contextes où la question de représentation orthogonalisée (disentangling) est essentielle.

Les performances sont démontrées d'une part sur des benchmarks standard en apprentissage et d'autre part sur des ensembles de données physiques confirmant la pertinence et la polyvalence de la solution proposée. Contrairement aux travaux précédents, nous montrons que, lorsque la détection est le but, il n'est pas nécessaire d'ajouter une configuration adversariale. Nous montrons aussi qu'une architecture conditionnelle et des métriques personnalisées sont indispensables.

Bien que les études de cas et les ensembles de données présentés dans cette thèse soient de caractéristiques de la physique, les résultats sont applicables à d'autres domaines, comme la surveillance des unités industrielles traditionnelles, car les méthodes génériques développées minimisent l'influence des spécificités de l'application.

Les résultats expérimentaux ont été très bien accueillis par les experts du domaine. Certains des modèles proposés ont déjà été intégrés et déployés dans l'infrastructure CMS de production. Une généralisation des stratégies proposées tout au long de cette thèse ouvre la voie à une automatisation complète de l'évaluation de la qualité des expériences de physique des hautes énergies.

Acknowledgements

The work presented in this dissertation would not have been attainable without the advice, guidance, and support of many who accompanied me over its course.

First and foremost, I direct my sincere appreciation to professor Cécile Germain. She took the risk of completing this project with me and offered her experience, patience, and attention. I extend my gratitude to my CERN supervisor, Dr Gianluca Cerminara, whose useful advice saved me from countless problems and research dead-ends. Dr Maurizio Pierini encouraged me to delve into this project and motivated me throughout, which I dearly appreciate. I thank my students Agrima Seth, Aman Hussein, and Mantas Petrikas I have had the pleasure of supervising, and from whom I learned a lot. My colleagues I have had worked with: Dr Virginia Azzolini, Dr Andrea Bocci, Dr Federico De Guio, Dr Giovanni Franzoini, Dr Francesco Fiori, Tomasz Krzyżek and Filip Široký. My colleagues from Université Paris-Saclay, Victor Estrade and Victor Berger for many discussions, suggestions and for taking care of me whenever I was in Paris. I thank Dr Danilo Rezende for taking his time and giving comments on my work. I appreciate Olmo Cerri for teaching me the basics. I am grateful for the CMS collaboration providing the data sets used in this thesis. I acknowledge all the members of the CMS Physics Performance and Dataset and the CMS DT Detector Performance Group I have had discussions with. Finally, I acknowledge the support of the CMS CERN group for providing the computing resources, especially Dr Felice Pantaleo.

I also send my love to the ones who kept me sane over these years and offered endless support and tremendous encouragement. My family and friends in Genéve, León, London, Łódź and Paris. I acknowledge my flatmates at the famous Emilia Noyères, without whom this project would have been concluded much earlier.

This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement 772369). CERN OpenLab sponsored Agrima's and Aman's internship at CERN, as part of the CERN OpenLab summer student program.

Contents

CHAPTER 1

Introduction

High Energy Physics (HEP) experiments involve large and complex detection apparatuses, which need to be operated with high availability to profit at best of the expensive beam-time provided by modern particle accelerators. Moreover, HEP data analysis relies on well understood experimental equipment to chase statistically rare phenomena. These requirements call for constant and reliable monitoring to spot potential malfunctioning. For this reason, all the HEP collaborations build complex Data Quality Monitoring (DQM) infrastructures and procedures, targeting prompt identification of anomalies arising from either hardware problems or data processing. The Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider (LHC) at European Organization for Nuclear Research (CERN) bases its quality assessment procedures on the scrutiny of a large number of statistical tests by detector experts comparing data distributions with the ones derived from good-quality references.

This method has proven to be effective in pinpointing problems and anomalies for over a decade, but it is swiftly reaching its limits. The procedure relies on the availability of a large number of highly-trained field specialists. Improved operation efficiency, together with a reduction in person-power costs, call for automation of the procedure. Moreover, the LHC experiments are being upgraded to cope with even larger data volumes and to use more complex hardware components. The effort to anticipate all possible types of failures with expert-based rule systems and statistical profiling using histograms, along with the increase of workforce needed for daily monitoring shifts, poses scalability concerns.

This thesis proposes a possible approach to the automation of Anomaly Detection (AD) in HEP, taking into account the domain-specific requirements and constraints. The HEP monitoring problems are usually challenging for conventional statistical-based methods due to their inherently multidimensional nature. Machine Learning (ML), and in particular, the sub-field of AD, must be summoned to automate the decision process. Besides, the HEP community expresses increasing interest in ML methods because of their applicability beyond quality control. Successful integration of ML within the DQM framework can serve as a showcase for application to other aspects of the HEP field with more stringent integration criteria, e.g. event selection processes.

Since the number of monitored quantities and corresponding failure modes is vast and no general solution is possible, the present work concentrates on addressing some of the most frequent monitoring patterns in the CMS DQM.

- Deep Neural Networks (DNNs) are used to analyze images representing geographically organized and high dimensional data. The study serves as an initial benchmark for comparing the Deep Learning (DL) based AD with the classical one and examining relevant aspects of ML, i.e. active learning and result verification. The experiments confirm the applicability of supervised DL in cases of a limited number of labelled samples as unprecedented detection efficiency is obtained. Additionally, leveraging both labelled and unlabelled data allows for training more sophisticated models for the detection of unforeseen failure modes.
- Deep autoencoders are used for quality control of the LHC collision data represented as multidimensional distributions. Semi-supervised and supervised approaches are examined, demonstrating the potential of the semi-supervised models in highlighting emerging anomalies. The experiments confirm the power of autoencoders in the context of novelty detection and explore their usage to ease the interpretability of the results, pinpointing the problematic features.
- Conditional Variational Autoencoders (CVAEs) are proposed for AD on hierarchically structured data. The study proposes a method addressing the still open ML problem of utilizing the rapid advancements in Variational Autoencoder (VAE) research for AD purposes in the realm of disentanglement techniques. The performance is demonstrated on standard ML benchmarks and physics data sets confirming the appropriateness and versatility of the proposed solution. In contrast with previous works done on CVAEs, it is found out that, when AD is the goal, an adversarial setup is not necessary. However, the original conditional architecture and tailored metrics are required.

While the case studies and data sets presented in this thesis are HEP specific, the findings in terms of AD models are applicable to other realms, like monitoring of traditional industrial units, as the core methods try to minimize the influence of domain-specific nuances.

This thesis is structured as follows. Chapter 2 reviews the CMS experiment design in the context of HEP science with a focus on the technical aspects relevant to the following chapters. Chapter 3 provides background material for ML AD focusing on the difference between classical approaches and the new, DL-based ones with an emphasis on the Variational Inference (VI). Chapter 4 considers the problem of detecting anomalies based on occupancy plots, an image-like representation of detector data often encountered in *quasi-real-time* DQM. Chapter 5 extends the methods proposed in Chapter 4 to applications in *post-mortem* DQM in the context of novelty detection on numerical data. Chapter 6 provides background and experimental results for a novel AD method proposed in the context of the monitoring hierarchically structured data sets. Chapter 7 delivers concluding remarks and future directions.

CHAPTER 2

Data Quality Monitoring at Compact Muon Solenoid Experiment

This chapter introduces the basic concepts and the framework necessary to make the following case studies understandable to readers with no HEP background. Even though the methods proposed in this work are designed to be domain-independent, the case studies have auxiliary domain-specific challenges that are formulated here, for completeness and clarity. This chapter also covers the motivations and goals of particle physics and HEP experiments, the technical design of the detectors, details, and reasons for robust DQM and the future challenges of the monitoring paradigm.

2.1 The Particle Physics Landscape

Particle physics research examines the nature of the fundamental constituents of matter and their interactions. The Standard Model (SM) of particle physics embodies the current understanding of physics, described for instance by Ho-Kim and Pham (2013). Figure 2.1 summarizes the known fundamental particles and their main properties.

According to the SM, fermions are the building blocks of matter. Arranged in three families, they interact through forces, mediated through carrier particles (bosons). Fermions are classified into leptons (*electrons*, *muons*, and *taus* and chain *neutrinos* that include ν_e, ν_μ and ν_τ) and quarks (*up*, *down*, *charm*, *strange*, *top*, *bottom*). These particles have associated antiparticles of the same mass but opposite quantum numbers (e.g. the electric charge). The SM unifies three fundamental interactions: the strong (mediated by *gluons*), the weak (mediated by Z and W bosons) and the electromagnetic (mediated by *photons*) force. Quarks interact via all three forces, leptons are not sensitive to the strong force, and neutrinos are subject only to the weak force. The SM does not account for gravitational force, whose effects are in any case negligible at the energy scale typically considered in HEP experiments.

Physics experiments are the way to validate the models and the theories describing our current understanding of the universe. Many, including Craig (2013), regard the SM as a

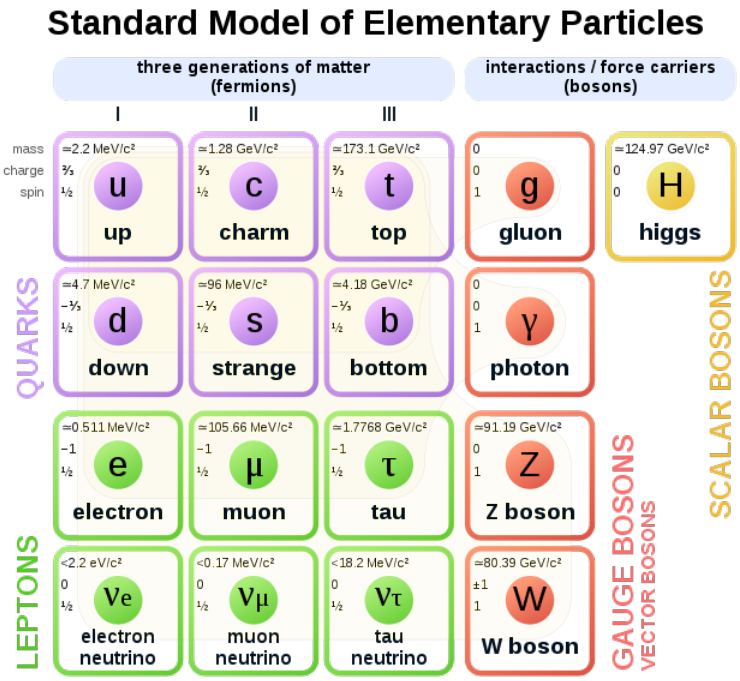


FIGURE 2.1: The SM of elementary particles: the 12 fundamental fermions and five bosons. From MissMJ (2006).

complete theory both from an experimental and theoretical point of view. Numerous attempts found no inconsistencies between the theory and the experiments within their level of precision and accuracy.

The SM was proven particularly successful in anticipating the existence of previously undiscovered particles. For instance, in 1964 Robert Brout, Francois Englert and Peter Higgs published a theory that explained the origin of mass through the so-called Brout-Englert-Higgs mechanism, described in Englert and Brout (1964) and Higgs (1964). It predicted a new carrier particle responsible for giving mass to the otherwise massless particles of the SM, a fundamental step to conciliate the SM with long-standing experimental findings (e.g. the mass of the electron). The particle, called the *Higgs* boson, was finally observed in 2012 at the CERN LHC, after decades of searches at other particle colliders.

Discovering Chatrchyan et al. (2012), and characterizing, Khachatryan et al. (2015), the Higgs boson was not the ultimate challenge particle physics is facing. The HEP field will continue to build complex experiments to examine more unsolved phenomena of the observed universe. A few examples of current research are listed below.

- In the early universe, the amount of matter and antimatter should have had been equal. However, matter constitutes almost the entire observable universe now. The observed differences in the amount of matter and antimatter are yet unexplained, see Bernreuther (2002).
- Based on the astronomical observations, galaxies are rotating with speed so high that the gravity generated by the observable matter could not hold them together, essentially tearing them apart. The unobservable part, referred to as dark matter, see Bertone

et al. (2004) for a general survey, which gives these galaxies extra mass and generates the extra gravity for them to stay intact is yet to be discovered.

- Neutrinos exhibit the properties of a particle as well as a wave. Each neutrino travels through space as a wave that has a different frequency. The flavour of a neutrino is determined as a superposition of the mass eigenstates. The type of flavour oscillates (see Gonzalez-Garcia et al. (2016)) because of the changing phase of the wave. The unsolved problems that remain are those of the Charge Conjugation Parity (CP) violation parameter, the neutrino mass hierarchy and the mass value of each neutrino.
- Gravity is described based on Albert Einstein's general theory of relativity, formulated within the framework of classical physics. However, the other types of fundamental forces are described within the framework of quantum mechanics and quantum field theory. Quantum gravity, see Rovelli (2011), is a theory that attempts to explain gravitational physics in terms of quantum mechanics. However, it still has to be examined experimentally.

Although not all particle physics experiments operate at a high energy scale, the questions listed above, among many others, remain open and are currently the main research focus in HEP. Since the 1950s, particle colliders represent the primary tool for this field of research. These machines are used to speed up and increase the energy of a beam of particles by generating electric fields that accelerate the particles and magnetic fields that steer and focus them. The accelerated particles are then made collide, and dedicated detectors are used to record the outcome of these collisions. As illustrated by Einstein's famous equation, $E = mc^2$, the kinetic energy of the colliding beams can be used to produce particles that are more massive than the accelerated ones. Following this principle, the history of HEP has been characterized by a continuous strive to accelerate particles to higher and higher energy, thus having access to more massive particles as a result of the collisions. The last step in this long-lasting journey towards the so-called *high-energy frontier* is the LHC, which started operating at CERN in 2008 and which gives access to higher energy levels than ever before, in the hope of unveiling not yet observed phenomena. Besides the push for higher energy, accelerator scientists are also pushing forward the limit of the intensity of the accelerated beams packing more and more particles in each collision. The higher the number of particles that physicists can squeeze into a beam, the more chances they have to catch a glimpse of rare processes to happen.

The intensity of the collisions comes with the drawback of requiring more and more precise detectors. In particular, the growing intensity also increases the probability of less interesting phenomena to happen, and the experiments need to efficiently discriminate the processes which could unveil new physics from those that are now considered well understood.

To achieve that the experiments require the most sophisticated solutions built by humankind and scientific coordination on an unprecedented scale. This complexity imposes stringent requirements on the data acquisition chain, which has to cope with an enormous quantity of data. Thus the HEP field calls for novel solutions to be explored and incorporated, including a more robust and automated monitoring scheme using the latest advances in Artificial Intelligence (AI).

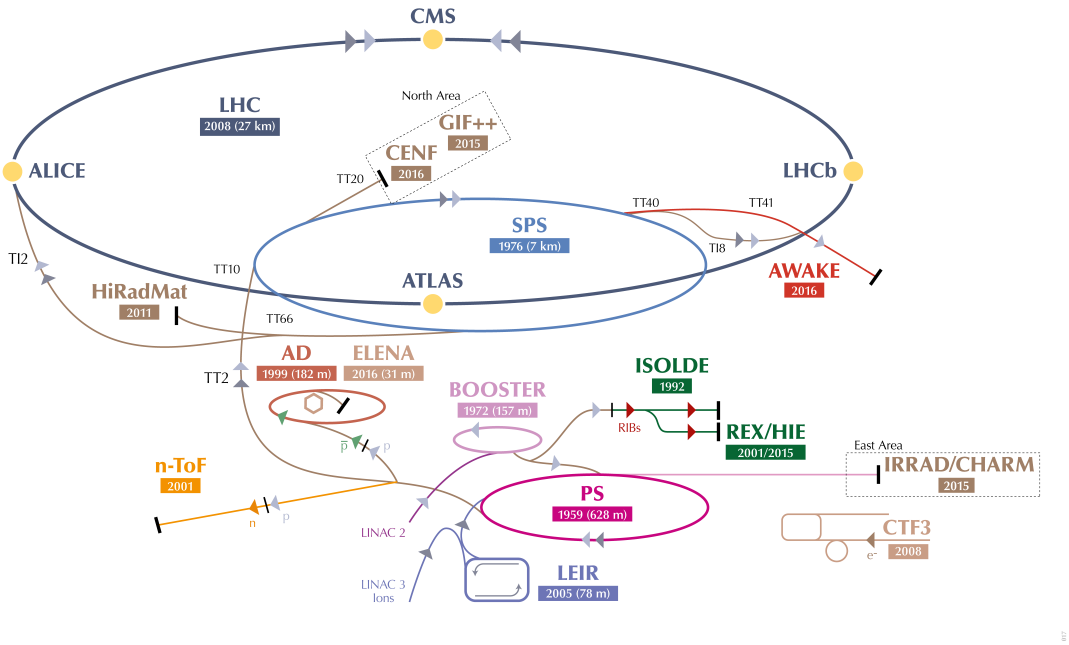


FIGURE 2.2: A view on CERN's accelerator complex. From CERN (2016).

2.2 The CERN Large Hadron Collider

As presented in the previous section, the HEP science needs a powerful accelerator and a high rate of collisions to produce sufficiently many rare events to be able to observe them. This section describes how the CERN LHC achieves these two goals, together with the trade-offs incurred by reaching unprecedented energy and collision rates. It briefly sketches what the LHC experiments are and how they integrate with the LHC infrastructure. Most of the technical details given here are relevant to the next chapters. The others must still be explained, to give a realistic picture of how complex the LHC operations are. Generally, the monitoring of the quality of the observational apparatus response to the collisions needs to be aware of these complexities.

The accelerator complex at CERN is shown in Figure 2.2. It consists of the LHC and a set of smaller accelerators, used for accelerating protons to higher energies. Each machine injects the particle beam into the next one, taking over the acceleration. The CERN LHC described in The LHC Study Group (1995) is the last element of this chain and hosted inside a tunnel that is approximately 26.7 km long. Protons in the LHC move at 0.999999991 times the speed of light at top energy, providing total collision energy of 13 TeV, making it the most powerful particle accelerator in the world. The construction of the LHC took ten years and happened between 1998 and 2008.

The beam injection starts with the hydrogen atoms which are taken from a bottle containing hydrogen gas and stripped from electrons to get protons. The protons are then injected into the PS Booster (PSB) at the energy of 50 MeV from Linac2 linear accelerator. The PSB accelerates the beam to 1.4 GeV, the Proton Synchrotron (PS) to 25 GeV and the Supper Proton Synchrotron (SPS) to 450 GeV. Protons are finally transferred to the LHC in both directions where they are accelerated to 6.5 TeV. The circular colliders offer higher collision energy as opposed to fixed-target energy experiments and a thus higher probability of generating rare decays.

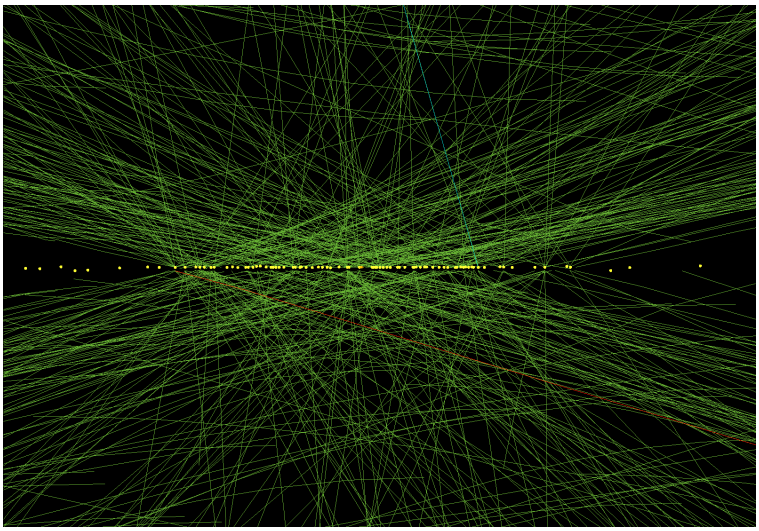


FIGURE 2.3: The spacing of 78 (PU= 78) collisions (yellow dots) at the CMS detector along the z-axis. The green lines represent tracks of the resulting particles. Credit: Andre Holzner.

The protons of the LHC circulate the ring in distinct groups (*bunches*). Under nominal operating conditions, each beam has 2808 bunches with each bunch containing about $1.15 \cdot 10^{11}$ protons. The bunch size is not constant around the ring as it gets squeezed (focused) as much as possible around the interaction points to increase the probability of collision. The bunch spacing is 25 ns or 7 m, yielding collisions at the remarkable rate of 40 MHz. The energy density and temperature produced in the LHC collisions are similar to those that existed a few moments after the Big Bang. In this way, physicists hope to understand better how the universe evolved.

Each injection of protons into LHC is referred to as a *fill*. The duration of one fill is not fixed under normal operating conditions and can vary between a few hours and a day. The *instantaneous luminosity* is a measure for the intensity of the collisions being proportional to the potential yield of a given physics process in every bunch crossing. It depends on the number of protons per bunch, the number of circulating bunches, the revolving frequency and the Gaussian transverse beam profiles in the horizontal and vertical directions. Measuring this quantity is critical for the operation of the LHC and its experiments.

Physicists refer to each bunch crossing as an *event*. The complexity of an event (i.e. number of particles produced) depends on beam intensity (also referred to as *luminosity*). The luminosity varies during each fill, resulting in a varying number of proton-proton interactions in the same event, referred to as *pile-up (PU)*. High luminosity is desirable from the scientific point of view since it increases the probability of rare processes to be produced. However, high PU causes technological challenges (as explained in Sections 2.3.2 and 2.3.3) related to the tighter spacing of the collisions (see Figure 2.3).

The luminosity, and as a consequence the PU, decreases during the fill as protons get consumed by the collisions. However, adjustments of the LHC beam-optics can result in discrete discontinuities and sudden increases of luminosity. Figure 2.4 shows the evolution of a typical LHC fill. In other words, the LHC conditions are not stationary in time. This variability poses an additional challenge for any ML-based solution for monitoring detector or physics quantities.

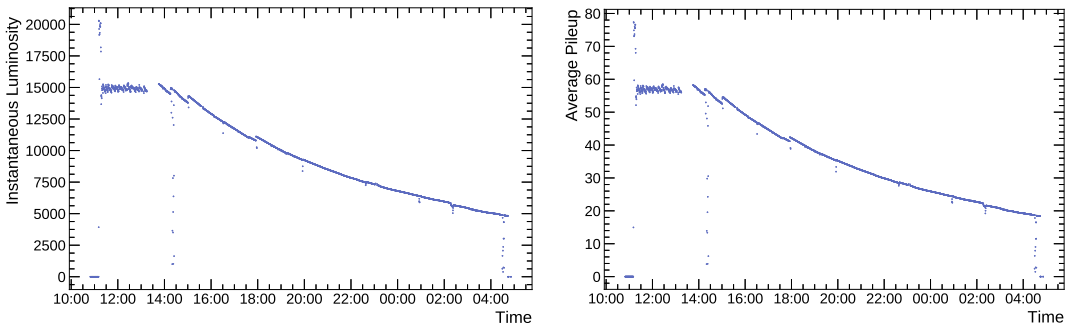


FIGURE 2.4: Evolution of the fill 6346 of LHC Run 2 at the CMS interaction point. The LHC makes adjustments during each fill. As a result, the instantaneous luminosity (left) and average PU (right) can increase by small steps. In this example, the downward fluctuations around 14:00 and 4:00 are due to routine operations on the beams. Smaller fluctuations and steps every 1-2 h are due to the optimization of the beam parameters. The jagged beginning is the so-called *lumi-levelling*, where the machine adjusts its configuration every few minutes to keep a stable PU. The spike at the beginning is before obtaining stable beams. A similar pattern applies to all the LHC fills.

Colliding the beams in the LHC is the responsibility of CERN. Exploiting these collisions is the responsibility of immense international scientific collaborations, the *experiments* in HEP jargon. Each experiment is associated with a *detector* which is a sizeable experimental apparatus that records and filters the byproducts of collisions and saves it in numerical form. The experiments are in charge of first designing and building their detector, then operating and monitoring it. Typically, the timescale of the experiments spans decades. For instance, in the case of the CMS experiment, the Letter of Intent was completed on the 1st of October 1992, the construction began in 1998 and completed in 2004[1]. Only in 2009 CMS registered first collisions. The experiment will continue to operate for the next 20 years.

The detectors are located in underground caverns, in four points where the two LHC beams intersect. These main LHC experiments are A Toroidal LHC ApparatuS (ATLAS), A Large Ion Collider Experiment (ALICE), CMS and Large Hadron Collider beauty (LHCb). Besides protons, the LHC can also accelerate Lead ions. ALICE, described in Aamodt et al. (2008), is a detector specialized in analyzing those types of collisions. The experiment studies the properties of quark-gluon plasma, a state of matter where quarks and gluons under conditions of very high temperatures and densities are no longer confined inside hadrons that provide a window onto the state of matter that existed in the early universe. The LHCb, described in LHCb Collaboration (2008), studies the properties of antimatter, specializing in the study of the slight asymmetry between matter and antimatter present in interactions of B-particles (containing b-quarks). While those two experiments study specific aspects of the LHC physics program, ATLAS, described in ATLAS Collaboration (2008), and CMS are general-purpose detectors, which are built using different technical solutions and design. They both cover the broadest possible range of physics at the LHC from precision measurements of the Higgs boson to searches for new physics beyond the SM. The CMS detector, shown in Figure 2.5, is located at Point 5 of the LHC, near Cessy, France.

[1]The Electromagnetic Calorimeter (ECAL) endcap was installed and calibrated just before the first LHC beams.

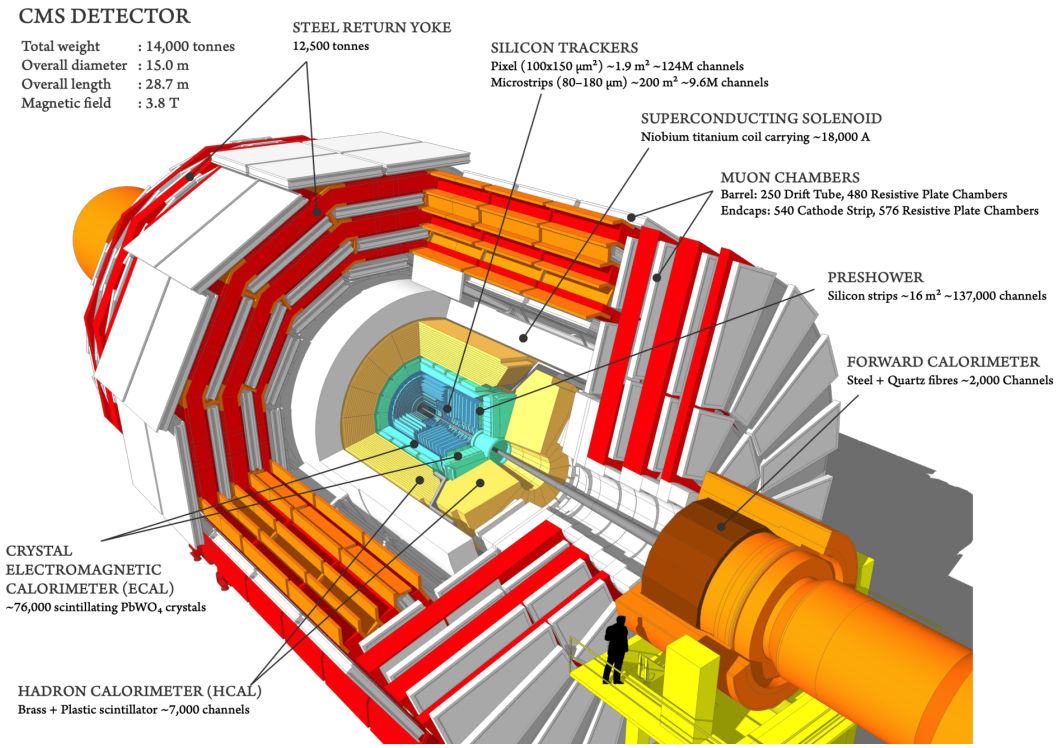


FIGURE 2.5: A cutaway diagram of the CMS detector with elementary information about different sub-detectors. From Marcastel (2013).

2.3 The Compact Muon Solenoid Experiment

This section introduces the main aspects of the CMS experiment, which are relevant to the case studies presented in this dissertation. Following the general introduction to the experiment, the section sketches the architecture of the detector and the specific role of its components. Finally, it explains how the enormous volume of data produced by collisions is handled with particular emphasis to the data-reduction and selection performed online by the so-called *trigger system*.

2.3.1 The HEP Experimental Data Pipeline

The CMS experiment can be used to illustrate the paradigm of HEP experiments. It is designed to cover a broad range of processes within the LHC physics program. Despite the theoretical differences between experiments, aspects of physics are probed similarly.

The essential steps for analyzing the data collected from all the detectors are similar. The first step characterizes all the different particles that were produced in each collision to reconstruct the process in full. This step is an arduous task for two reasons. Firstly, as explained before (Section 2.1), most of the resulting particles are unstable and *decay* quickly into a cascade of lighter particles. The particles of interest are observed only indirectly, by the final decays products. Secondly, these products are observed only through their interactions with the detector and subject to measurement uncertainty. Overall, the role of the CMS detector is to collect raw data about the interaction of these so-called *secondary particles* with the

sensitive layers in order to reconstruct their *momentum* and *energy*. The properties of the decayed parent particle are inferred from these quantities, and the inference chain is continued until reaching the most massive *primary particles*. At an intermediate step, a trigger system discards the vast majority of bunch collisions, which contain uninteresting events (see Section 2.3.3). Each retained event contains only a few particles of interest, reconstructed from hundreds of low-level signals. Each event is characterized by many measurements, along with many additional, high-level features that have been engineered by physicists. The final analysis interprets vast collections of such events, see Khachatryan et al. (2015) or Adam-Bourdarios et al. (2014).

2.3.2 The Detector

This section specifies the CMS components directly related to the monitoring application subject of this dissertation: the muon Drift Tube (DT) detector and the trigger system. The rest of the technical details are here for completeness.

The CMS acronym was not chosen by coincidence. The detector is *Compact*, weighing more than all other LHC experiments while occupying less volume than the other multi-purpose detector at the LHC: the ATLAS detector. The *Muon* word is a reference to its robust and precise muon spectrometer. *Solenoid* refers to the configuration of the magnetic field produced by a large and powerful superconducting magnet. A detailed description of the detector is in Chatrchyan et al. (2008), together with a definition of the used coordinate system and the relevant kinematic variables.

The CMS detector has a cylindrical structure. The apparatus is 28.7 m long, 15 m in diameter and weighs $1.4 \cdot 10^4$ t, i.e. twice the weight of the Eiffel Tower, see Hanser (2006). Monitoring the proper functioning of the system is possible only because of the essential repetitiveness of its structure. The detectors have many layers (or *sub-detectors*) that serve a particular role in the reconstruction of collisions and employ specific technologies as sketched in Figure 2.6. As mentioned before, when crossing the detector, a particle produces only two kinds of raw data: a *tracking hit* and an *energy deposit*. Physicists' goal is to track and characterize all the different particles that were produced in each collision. Recorded trajectory and energy allows for measurement of particle positions in space, their charge, mass, and momentum.

Tracking is the act of measuring the trajectory of charged particles. Modern tracking devices reveal the nodes of charged particles through electrical signals. Particles that carry an electromagnetic charge and traversing a thin layer of material release an electronic signal by ionizing the gas in the detector (a *hit*). The electrons produced by the detector material ionization are collected by the readout electronics in a region close to the track trajectory. If the material is segmented in small active regions (*pixels* and *strips*), the crossing point can be measured with high precision. Due to the presence of a magnetic field, charged particle trajectories are bent along a helix. Based on the result of software tracking algorithms connecting the points the particle has traversed, a circle of a radius R in the transverse plane is measured. The radius R and the momentum are proportional ($R \sim \frac{p}{m}$) and thus both could be determined. Higher momentum produces straight lines, while low creates tight spirals.

Two specialized types of tracking devices are the inner tracker (vertex and strip detectors) and muon chambers. Vertex detectors are located in the innermost part of the detector, close

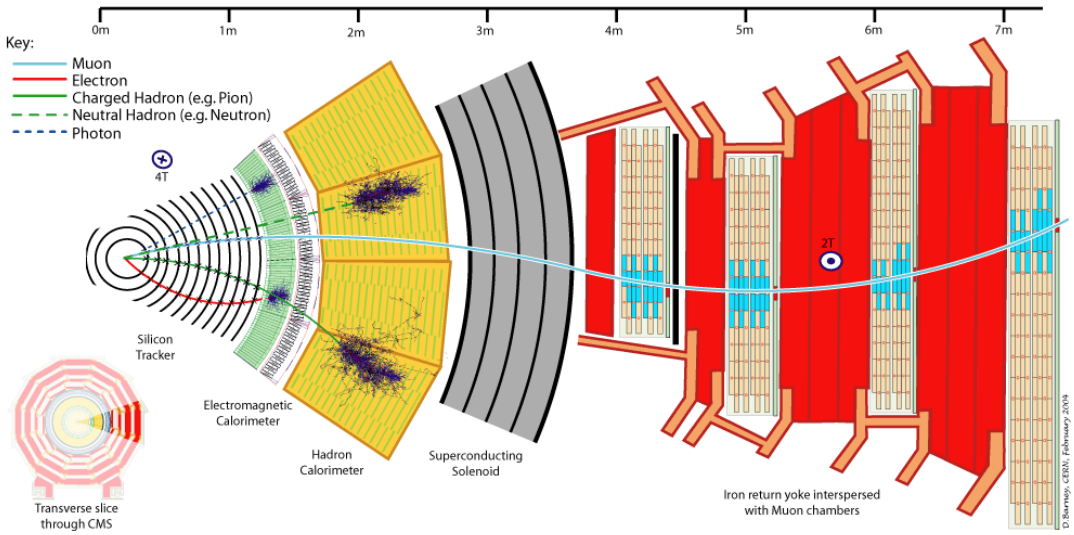


FIGURE 2.6: A sketch of the particle interactions in a transverse slice of the CMS detector, from the collision point to the muon detectors. From Davis (2016).

to the interaction point. Muon chambers are located at the outer layers of a detector assembly as muons are the only charged particles able to travel through the meters of dense material. In CMS, the silicon tracking devices (described in Karimäki et al. (1997); CMS Collaboration (2000a)), the Pixel (seen in Figure 2.7) and the Strip detectors, are located in the proximity of the interaction point. The Pixel at $r < 15$ cm with sensors containing a vast number of pads has a total of 76 million readout channels. The Strip detector located at $r > 0.2$ m completes the design. In CMS, muons are measured with detection planes instrumented with four detector technologies: DT, cathode strip chambers, resistive plate chambers, and gas electron multipliers. A detailed description of all the CMS muon detectors can be found in Sirunyan et al. (2018). Finally, the 3.8 T solenoidal magnetic field at CMS, critical for measuring the trajectories, is produced by NbTi superconducting magnet CMS Collaboration (1997c).

Calorimeters are devices that measure the energy of the incoming particles. Neutral particles such as photons and neutrinos are not visible in tracking devices, but the energy they deposit in the calorimeters reveals them. The CMS detector includes two types of calorimeters, described in CMS Collaboration (1997a) and CMS Collaboration (1997b): ECAL and Hadron Calorimeter (HCAL). They are made of different materials and use different measuring technologies. ECAL is made of 76000 scintillating $PbWO_4$ crystals, HCAL is made of brass and plastic scintillators. The ECAL absorbs electrons and photons. Strongly interacting particles (hadrons) begin to lose energy in the ECAL but are only stopped in the HCAL.

The Drift Tube Muon Detector

An illustration of the internal structure of a DT chamber is shown in Figure 2.8. Each chamber, on average 2×2.5 m in size, consists of 12 layers of DTs. Layers are arranged in three groups of four. Each layer contains a variable number of tubes depending on the position in the detector, up to 96. The middle group measures the coordinate along the direction parallel to the beam and the two outer groups measure the perpendicular coordinate. Each tube

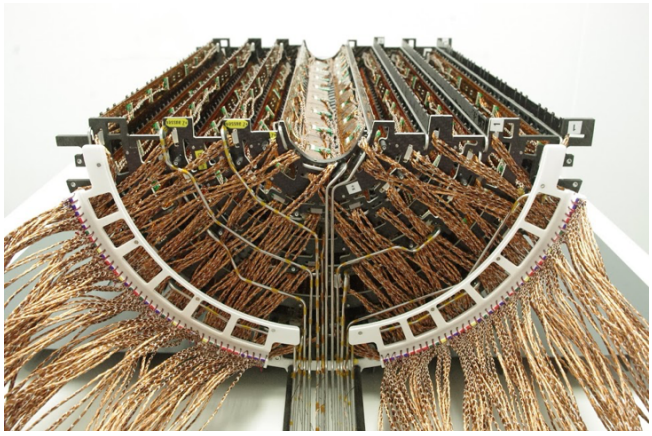


FIGURE 2.7: A barrel pixel detector halves, assembled at Paul Scherrer Institute, with modules made by several European consortia. The photograph shows very lightweight mechanical support, CO_2 cooling tubes and an impressive amount of cabling in a tight volume. From Paul Scherrer Institute (2017); Gill and EP-CMX (2017).

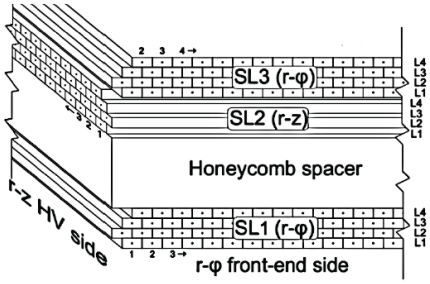


FIGURE 2.8: Schematic view of the one DT chamber showing the position and orientation of the tubes. From CMS Collaboration (2010).

corresponds to one readout channel (briefly referred to as a *channel*). By combining the information provided by the channels, one can determine the trajectory of the particle crossing the chamber. The iron return yoke captures the flux lines of the solenoid magnetic field.

The chamber numbering schema follows that of the iron of the yoke, consisting of five wheels (see Figure 2.9) along the z-axis, each one divided into 12 azimuthal sectors (see Figure 2.10). The wheels are numbered from -2 to +2, sorted according to global CMS z-axis, with wheel 0 situated in the central region around the proton-proton collision point. The sector numbering is assigned in an anti-clockwise sense when looking at the detector from the positive z-axis, starting from the vertically-oriented sector on the positive-x side in the CMS coordinate system (sector 1). Chambers are arranged in four stations at different radii, named MB1, MB2, MB3, and MB4. The first and the fourth stations are mounted on the inner and outer face of the yoke respectively; the remaining two are located in slots within the iron. Each station consists of 12 chambers (one per sector) except for MB4 (which contains 14 chambers). The total number of chambers is then $5 \times (3 \times 12 + 14) = 250$.

2.3.3 The Trigger System

Much less exposed than the detector, the trigger system is nonetheless an equally essential part of the CMS acquisition process. Due to the intrinsic statistical nature of the particle

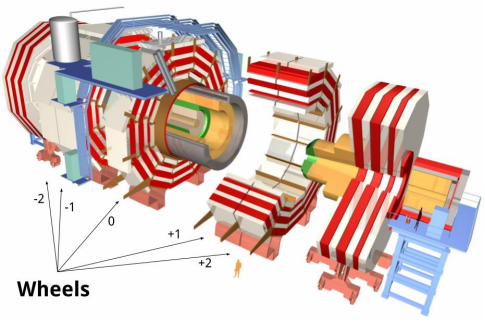


FIGURE 2.9: Magnified view of the CMS detector showing the wheel structure. The white volumes represent the muon chambers while the red volumes represent the iron return yoke.

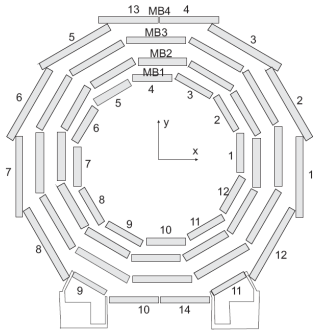


FIGURE 2.10: Numbering schema of the sector and stations of DT chambers in one wheel. From CMS Collaboration (2010).

physics phenomena, the HEP experiment needs to collect a large amount of data with rare events. The LHC operates at a remarkable rate of 40 million events per second. Each CMS event corresponds to around 1 MB of data in unprocessed form. Thus the volume of the data produced in collisions results in hundreds of Exabytes (EBs) of data per year, making the LHC one of the largest sources of data in the world today. For instance, considering the eight months of the data-taking period in a solar year and a duty cycle of 60%, the LHC delivers more than 400 EBs. Ideally, physicists would like to keep all these raw data. However, it would often be useless and above all, technologically impossible. Useless, as most of the events can be discarded upfront as they follow a known process for the current state of physics. Impossible, as at present day it is not possible to even transfer most of the data from the detectors to the offline data facilities due to network constraints, much less to permanently store it for future processing. Due to these understandable storage constraints and other technological limitations (e.g. fast enough readout electronics), the experiment is required to reduce the number of recorded data from 40 million to 1000 events per second. To this purpose, a hierarchical set of algorithms collectively referred to as the *trigger system*, is used to process and filter the incoming data stream. Trigger algorithms, described in Khachatryan et al. (2017), which are the start of the physics event selection process, are designed to reduce the event rate while preserving the physics reach of the experiment. The CMS trigger system is structured in two stages of selection using increasingly complex information and more refined algorithms.

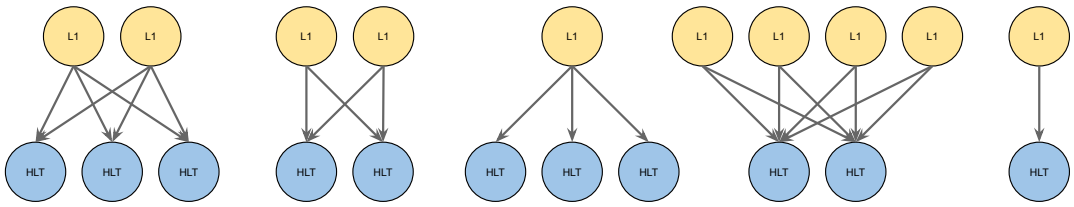


FIGURE 2.11: Simplified, schematic graph inspired by the trigger system configuration. Blue nodes represent HLT nodes, while yellow L1 Trigger nodes. Each link is unidirectional starting from yellow nodes. For every fill, the graph has a few hundred nodes spread approximately equally between HLT and L1 Trigger nodes. The connection between L1 Trigger and HLT nodes can be seen as a hierarchical directional graph from the L1 Trigger to the HLT system.

- Level 1 Trigger (L1 Trigger), see CMS Collaboration (2000b), implemented on custom-designed electronics, reduces the 40 MHz input to a 100 kHz rate. L1 Trigger algorithms are implemented on Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASICs) and have 3.2 μs to decide whether to pass an event to the next step. The algorithms are based on local information from sub-detector components.
- High Level Trigger (HLT), see CMS Collaboration (2002), is a collision reconstruction software running on a computer farm of about 32 thousand commercial processors. It scales the 100 kHz stream from the L1 Trigger rate down to 1 kHz. The HLT system runs a faster and coarser version of the reconstruction software used for offline reconstruction and analysis. It uses global detector information, takes advantage of regions of interest to speed up the reconstruction and rejects events as early as possible. The HLT decision has to happen in 100 ms on average.

To better illustrate the challenge, at the luminosity at which the LHC currently operates, out of the 40 MHz of collisions, the rate of production of Higgs bosons is about 0.4 Hz. The trigger system needs to be able to identify the events efficiently in the limited time available.

Both L1 Trigger and HLT systems implement a set of rules to perform the selection (called *trigger nodes* or *paths*). A trigger node corresponds to an algorithm that probes a specific pattern (*signature*) in the event or looks for specific physics objects. There are nodes dedicated to the primary particles, e.g. electrons, muons, or hadronic energy in the calorimeters. In general, physics analyses looking for sophisticated and rare signatures can design their selection to enrich the data of rare events. At present, the HLT runs about 600 of these independent selection algorithms. In the spirit of refining the selection in consecutive steps, each HLT node is seeded in input by the events selected by a configurable set of L1 Trigger nodes, see Figure 2.11.

2.3.4 Data Organization

The CMS data are organized in acquisition runs (or just *runs* in CMS jargon), not to be confused with LHC Runs which are years long. They correspond to a given setup both of the CMS detector and the LHC accelerator. Their duration is varying from as little as a few minutes to as much as $\sim$ 20 hours. Each run is divided into Lumisections (LSs), a time interval

corresponding to a fixed number of proton-beam orbits (2^{18}) in the LHC. It amounts to approximately 23.31 s, time large enough to measure the average instantaneous luminosity (see Section 2.2), monitor the status of the sub-detectors (see Section 2.4) and small enough to be used as the atomic of the data for physics analysis. Typically the analysis is not performed on a single event but on a collection of them grouped by given times when the detector response was accurate. Each event can be identified uniquely by specifying the event number, the LS number, and the run number.

2.4 The CMS Data Quality Monitoring Infrastructure

Failures in a large and complex apparatus like the CMS experiment are unavoidable. The certification of the CMS data as usable for physics analysis is a crucial task to ensure the quality of all physics results published by the CMS Collaboration. For this reason, the collaboration set up an integrated monitoring system, including processes, methods, and software infrastructure, referred to as the DQM. The stringent quality criteria ensure that physics analysis is performed on good-quality data only. The failures are quite frequent: overall, 7%[2]. of the detector components manifest problems, while 2% of the acquired data is discarded.

Moreover, this relatively high figure is mostly not due to significant, easily detectable, malfunctions of the detector as a whole, but to localized problems. In most cases, entirely relevant physics analysis can still be achieved on the data taken by the not-faulty parts of the detector. Thus, an equally critical goal of the monitoring system is to be as precise as possible in spotting the defects, not reporting only an overall status.

This section deliberately uses a very operational point of view to describe the DQM to highlight the potential benefits and the challenges of introducing some level of automatic decision making. Details on the methods and quality indicators currently in use are given in the next chapters.

2.4.1 Overview

Within the CMS collaboration, physics analyses are performed only on *good-quality* data that require prompt and accurate identification and flagging of the problematic data. Imposing quality criteria is performed by the two main domains of the monitoring chain.

- *Online monitoring* provides live feedback on the quality of the data while they are being acquired, allowing the operator crew to react to unforeseen issues identified by the monitoring application, further described in Section 2.4.2.
- *Offline monitoring* was designed to certify the quality of the data collected and stored on disk using centralized processing (referred to as the event reconstruction, that converts detector hits into a list of detected particles, each associated with energy and direction), further described in Section 2.4.3.

These two validation steps differ in three main aspects.

[2]Calculations are based on the DT sub-detector data.

- First, there is a difference in latency of the evaluation process. Online monitoring is required to identify anomalies in quasi-real-time to allow the operators to intervene promptly while the offline procedure has a typical timescale of several days.
- Second, the fraction of the data which they have access to varies. Online processing runs at a rate of 100 Hz, corresponding to approximately 10% of the data written to disk for analysis (in order not to flood the monitoring system). The offline processing takes as input the full set of events accepted by the trigger system ($\sim$ 1 kHz of data).
- The third is the granularity of the monitored detector components. While offline monitoring requires identifying the only overall status of the sub-detectors, online should determine faulty sub-detector elements.

Despite their specific characteristics, these two steps rely on the same AD strategy: the scrutiny of a long list of predefined statistical tests, selected to detect a set of known and possible failure modes. These statistical tests are presented as a set of multidimensional plots (*histograms* in HEP jargon) for experts' convenience. These histograms are monitored by detector experts, who compare each distribution to a corresponding reference, derived from good-quality data in line with predetermined validation guidelines. The experts look for unexpected effects that could affect analysis level quantities, e.g. noise spikes, dead areas of detector problematic calibrations.

A separate category of online monitoring is the Trigger Rate Monitoring (TRM), further described in Section 2.4.4.

Further details on the infrastructure used for CMS DQM are given in Schneider (2018). The central component of the DQM system of CMS is a web-based service for browsing data quality histograms, called DQM GUI, described in Tuura et al. (2010). Currently, the system has more than 50 TBs of monitoring data.

2.4.2 Online Data Monitoring

Online DQM samples events that are selected during the HLT processing and focuses on monitoring the status of the various sub-detector components and the LHC beam conditions in quasi-real-time. The histograms are generated with very low latency live monitoring of detector performance during data taking. The live display is updated every LS. Statistical tests are performed to compare these histograms to a set of predefined references, representing the typical detector response during normal operating conditions. That allows for efficient detector operation by giving feedback on its status to the experts and the operators handling the data-taking. Expert shifters acknowledge the alarms and may decide to intervene (up to stopping the data taking), using the histogram comparison and evaluating of the problem severity. The primary role of a shifter is monitoring the plots constantly and contacting the right sub-system expert when the problem occurs. The knowledge of the LHC running conditions and the history of possible issues identified in the past are vital ingredients in this decision process. At the end of each run, the shifter is asked to fill in a quality flag for each sub-detector. Sub-systems with apparent problems should be marked as *bad* unless the shift leader or the sub-system expert says otherwise.

This thesis explores ML applications to the online DQM procedure in Chapter 4.

2.4.3 Offline Data Certification

The data acquired by the experiment are scrutinized by a procedure called Data Certification (DC), which ensures they are usable for all physics analysis. This procedure is the last step of the DQM apparatus of the experiment and utilizing all the reconstructed events. The DC process ensures sufficient accuracy and clarity to maintain an excellent detector and operating efficiency. Experts are trained to discover problems and pinpoint errors in the detector hardware or reconstruction software based on the assessment of hundreds of histograms filled with specific critical quantities. Additional log messages (aggregated in Rapsevicius et al. (2011)) allow for fine-grained analysis of the acquired monitoring data. The final certification flag is attributed, again, by comparing results to a predefined reference, representing the typical detector response during normal operating conditions. Certification shifters evaluate problems using their knowledge of the history of possible issues identified in the past.

The monitoring data comes from different CMS sub-detectors, and the global quality of data depends on the combinatorial performance of each of them. Typically, experts are trained to assess specific sub-detector behaviour. That results in having up to seventy people involved in the process. Furthermore, the time constraints (data has to be certified as quickly as possible) and complexity of the decision implicates the human-driven process to be prone to mistakes and introducing some level of quality label contamination.

This thesis explores ML applications to the DC procedure in Chapter 5.

2.4.4 Trigger Rate Monitoring

As discussed in Section 2.3.3, the trigger system is implemented as a broad set of nodes, each selecting a particular physics signature. The amount of data accepted by the trigger system, the trigger rate, is briefly referred to as the *rate* in CMS jargon. This yield of each of the nodes is a critical quantity that needs to be continuously monitored. A rate lower than expected might hint a loss in efficiency of the particular selection. On the other hand, high rates might create problems for the data acquisition chain and very often are the symptom of detector instabilities. Thus, they often provide early hints of failures. As the LHC pushes to higher collision luminosity, CMS has to be ready to promptly respond to emergencies when the rates fail to stay in the expected range. The trigger operation group has developed a set of software tools and protocols to thoroughly monitor, describe and identify problems based on reported rates. The next paragraphs detail the pipeline of TRM software, described in Wightman et al. (2018).

The rate of the physics processes determining the trigger rate decreases with the luminosity and, as a consequence, with PU. Consequently, the recorded collision rates decrease as well as they primarily depend on the luminosity of the beams. In practice, TRM predicts an average rate per bunch-crossing as a function of an average measurement of the PU for each LS. These predictions are then compared to the recorded rates as data are being collected, spotting small and unexpected deviations. In Figure 2.12, the red lines correspond to those predictions, while the blue dots are the actual values readout by the monitoring. The model describing the expectation is derived from a best-fit approximation (i.e. fitting the rate values as a function of average PU) limited to linear, quadratic or exponential regression. These prediction models are generated ahead of time using past data, selected from

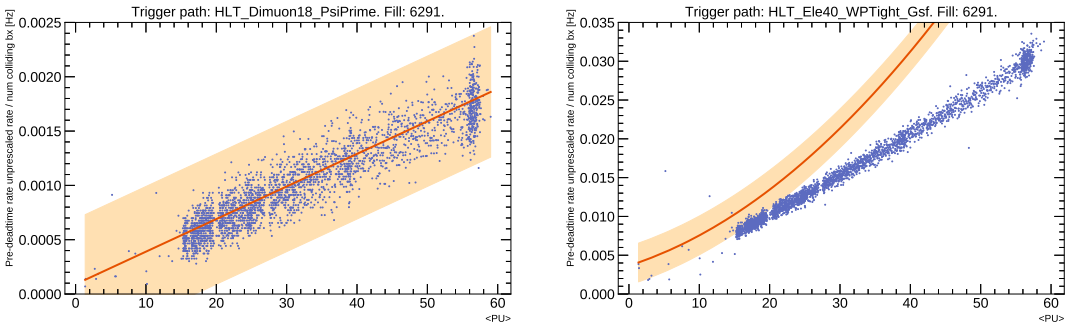


FIGURE 2.12: Observed rates as a function of average PU (blue dots), compared to the predicted dependence (red line) and its uncertainty (in the orange band) generated using TRM software. The plots above show an example of a well (left) and poorly (right) predicting model.

a list of known good-quality fills. Before fitting, raw rates are corrected for deadtime, random sampling suppression factors (*prescales*) that might have been applied and the number of colliding bunches in the LHC. Such preprocessing facilitates comparisons and extrapolations between different fills with different conditions. The final regression model is selected based on least-squares minimization, with a bias towards more straightforward (i.e. linear) fits and each trigger node is fitted independently from others. The models are updated periodically (approximately every other month) to account for changes, e.g. in the sub-detectors, trigger algorithms or calibration updates.

TRM software provides a list of 20 trigger nodes for real-time monitoring as a console-based application. The list is selected to cover CMS sub-detectors and physics objects (e.g. muons, photons, and so forth) and is monitored by the on-site shifter 24/7. The application is updated every 60 s, averaging readouts from the last three recorded LSs. The currently implemented strategy highlights trigger nodes when a deviation from the expected value is more than 5σ and alarms the monitoring crew. The script automatically sends an e-mail to trigger experts with information about these alarms and add an audio alarm that is raised if specific rates go above maximum acceptable thresholds, e.g. a total L1 Trigger Trigger rate above 100 kHz. If alarms persist over a long period, they are further evaluated. The shifter task is to determine if the genuine trigger system malfunction drives the alarm.

TRM also provides tools used for offline DC as trigger rate plots make it trivial to spot fills where a particular sub-detector was having problems. Rate versus PU plots (see examples in Figure 2.12) for all L1 Triggers and HLTs are integrated into the central CMS Web-Based Monitoring service, described in Soha (2011).

This thesis explores ML applications to the TRM procedure in Chapter 6.

2.5 Challenges and Long Term Plans

In 2013 the CERN Council adopted the new European strategy for Particle Physics, as summarized in Council (2013), setting priorities to exploit the full potential of the LHC, including high luminosity upgrade of the accelerator complex and the detectors. The upgrade aims at collecting ten times more data than in the initial design by around 2030, pushing instantaneous luminosity of the LHC to record high. As a result, the expected PU will reach 90

collisions per bunch crossing. The upgrade also includes replacing some of the detectors, e.g., the upgrade of the tacker detectors Tricomi (2014) or the calorimeter Magnan (2017). In particular, the trigger system will have to cope with higher rates, and the monitoring needs will become even more stringent.

The CMS Collaboration adopted the three-layer expert based monitoring protocol for LHC Run I (2010-2012) and in Run II (2015-2018). However, the ever-increasing detector complexity (especially in the context of high luminosity upgrade), monitoring data volumes and the necessity to cope with different LHC running scenarios call for an increasing level of automation of the process in the future. Already, the amount of histograms to monitor is challenging for a single shifter, while the number of histograms to monitor increases every time a new failure mode is identified and consequently added to the list of known potential problems. Furthermore, human intervention and currently implemented tests require collecting a substantial amount of data, implying a detection delay. Last but not least, the cost in terms of human resources is substantial, i.e. the 24/7 DQM shifter and the expert personnel responsible for updating the good data references and related instructions.

The only practical way to prepare this future challenge is to start developing and adopting automated AD in the current LHC Run. This early adoption will allow growing the expertise within the collaboration around the development and operation of various ML models and tools.

In this context, the work presented in this thesis represents the crucial step in a complete change of paradigm for the CMS DQM. The first structured effort is presented here to introduce advanced ML models for AD in the sophisticated monitoring infrastructure of the experiment.

2.6 CMS DQM and Machine Learning

ML methods open up the possibility of providing additional quality indicators in the current CMS DQM procedure as the decision function can be learned directly from the extensive archives of the past monitoring data and corresponding labels provided by detector experts. In the future, the monitoring system should take the burden of routine expert checks prefiltering the data and requiring expert judgment only in cases where the algorithmic decision is not clear. The load on the humans will substantially reduce (as discussed in Borisyak et al. (2017) in the context of the CMS DC).

It is possible to generalize and group the monitored quantities according to typical patterns. At the detector level, experts typically look at hit or occupancy maps, presenting relevant quantities in a geographical/topological fashion. These are further described in Sections 4.1 and 4.1.1. The purpose of such monitoring is to identify problematic regions in the detector. If such regions appear during the detector operation, the collaboration needs to know precisely when the problem appeared and how to intervene. Detecting anomalies on those maps is an image classification task: supervised for known problems and semi-supervised for unrecognized ones (Chapter 4). The DC step performs routine physics level checks on physics objects, i.e. hadrons, leptons, photons, and so forth when experts look for anomalies in the statistical distributions of fundamental physics quantities. While Chapter 4 experiments with novelty detection as a supplementary extension, Chapter 5 works exclusively on

this problem with higher-dimensional data. Finally, the correct interpretation of the monitoring data often requires prior knowledge of the configuration and causality relations of different detector components. Representation learning techniques open the way to the initial automation of this practice. Chapter 6 covers an application of this kind in the realm of the TRM.

The task of transitioning to ML-based monitoring is non-trivial. The high data dimensionality precludes simple parametric density estimation of the normal behaviour. Labelled instances are often not available for online monitoring, while offline monitoring the label contamination is difficult to estimate. Thus, even supervised techniques pose an implementation challenge. As the failure scenarios expand in quantity, and the conditions evolve with time, the solutions for model retraining must be implemented as well.

CHAPTER 3

Machine Learning Anomaly Detection

Chapter 2 presented a complicated task for experts to perform manually, especially in light of the long term plans of the HEP community. Instead of relying on expert knowledge, the protocol could be augmented with the help of recent progress in ML techniques. In this chapter, ML AD solutions are discussed in light of both the operational feasibility of monitoring practices and the *a priori* knowledge of the data.

As stated by Samuel (1967), *ML is the field of AI and computer science that gives computers the ability to learn without being explicitly programmed*. The majority of practical ML uses *supervised learning*. In such setup, given a data set space of observations $\mathcal{X}$ with input variables x and a label space $\mathcal{Y}$ with output variables y, the computer model (further briefly referred to as *model*) learns a function $h : \mathcal{X} \to \mathcal{Y}$ so that $h(x)$ predicts the output y. The target quantity y can be discrete or continuous, defining classification and regression tasks. In the absence of y, full or partial, the learning process is called *unsupervised* or *semi-supervised* respectively. Constructing good models from data sets boils down to selecting the hypothesis $h \in \mathcal{H}$ that minimizes (or maximizes) a particular objective. Historically $\mathcal{H}$ is called a hypothesis space. It is important to note that ML goes beyond simple function fitting as many auxiliary techniques are also included in this field, e.g. active learning, see Section 4.2.4.

A standard definition of an anomaly (or an outlier) is that it is an observation that appears to be inconsistent with the remainder of the data set, Grubbs (1969); Barnett and Lewis (1978). Anomalies can be caused by experimental or measurement errors but sometimes are indicative of a new, previously unknown, underlying process. In fact, Hawkins (1980) defines an outlier as *an observation that deviates so much from the other observations as to arouse suspicions that a different mechanism generated it*.

A common need when analyzing real-world data sets is determining which instances stand out as being dissimilar to all others, which is the goal of AD. Many applications require being able to decide whether a new observation belongs to the same distribution as existing observations (set of inliers) or should be considered as different (outliers). Often, this ability is used just to clean the data sets. Having identified the samples that contain errors, the next step is to fix the inconsistencies somehow or minimize their impact in the final model. In this context AD is just a preprocessing step. However, AD could be a stand-alone task. Detecting and analyzing anomalies reveals useful information about the characteristics of

the data generation process. Perturbations of healthy behaviour may indicate a presence of faults in the system, which makes AD particularly useful for monitoring tasks. Firstly, the ML AD can build the required detection model automatically based on some given training data. Secondly, it has the advantage of detecting previously unknown failures. That brings us to an important distinction.

- In the **novelty detection** context, the training data is not polluted by outliers, and the estimators detect previously unobserved patterns in new observations.
- In the **outlier detection** context, the training data is contaminated with outliers, and the estimators fit the regions where the training data is concentrated, ignoring the anomalous observations.

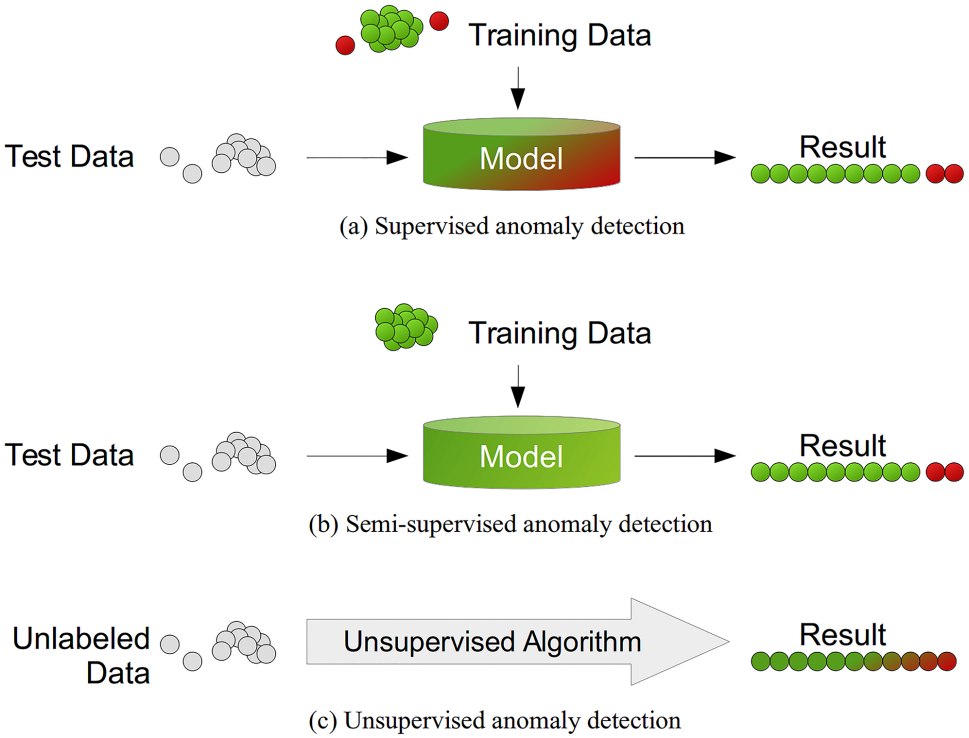
There are countless applications of AD such as credit card data fraud or identity theft detection, network traffic intrusion detection, medical imaging tumour detection, or HEP experiment sensors readings analysis (to determine fault in a component of the experiment) to name a few.

This chapter describes the different types of approaches to AD in Section 3.1, analysis algorithm performance evaluation in Section 3.2, overviews classical and popular techniques for AD in Section 3.3 addressing their limitations. Finally, it explores the new approaches based on DL architectures in Section 3.4 and in particular the VAE in Section 3.5.

3.1 Types of Anomaly Detection

Typically a detection model defines a region representing normal behaviour to declare anything outside that region as an anomaly. The amount of labelled data at hand plays a pivotal role in choosing the methodology of defining such regions. Based on the availability of the labels, indicating whether the sample is an inlier or an outlier, AD techniques operate in one of the following three approaches: *supervised*, *semi-supervised* and *unsupervised* (as illustrated in Figure 3.1).

The AD techniques usually assume rarity of abnormal events (considered as outliers concerning the normal generating process) and lack of a complete set of typical examples of all possible behaviours. Nonetheless, if the representative examples are available, the AD reduces to a case of binary classification (supervised learning), with possibly the help of various re-sampling methods, see Aggarwal (2014), or reformulation of the objective function for dealing with class imbalance, see Cowan et al. (2011). The class weight correction is critical as most of the classification techniques assume an approximately equal distribution of data classes and underperform when a class is severely under-sampled or utterly absent from the training set. A particular case of supervised AD is a Positive-Unlabelled Classification when a limited number of instances of the positive class are available with the unknown proportion of outliers in the negative set. Supervised AD constructs a predictive model for all classes (multiple in case of multiple anomaly types). Unseen data instance is compared against the model to determine the class it belongs to. In many application-specific scenarios, the supervised methods provide better detection rates than other approaches since they have access to more information. However, the primary technical issue, which makes supervised methods not suitable in many areas, is the shortage of available anomalous instances.



(a) Supervised anomaly detection

(b) Semi-supervised anomaly detection

(c) Unsupervised anomaly detection

FIGURE 3.1: Differences between supervised, semi-supervised and unsupervised AD approaches. From: Goldstein and Uchida (2016)

The model accuracy depends on the labelled training data, which in practice will not cover all possible types of anomalies. Besides, labelling is often an expensive, burdensome and time-consuming task (if not impossible at all) as experienced human annotators do it. If labelling is possible, accurate labels are usually not guaranteed and mislabelling may result in worse detection rates. A simple remedy may be an artificially generated example set, Theiler and Cai (2003); Abe et al. (2006). However, in most practical, real-life settings, the anomaly generating mechanism is non-existent, or the simulation assumptions may turn out to be too simple. The bottom line, a supervised AD may always serve as the first step of the AD pipeline as the examples of the previous anomalies can be used to focus further search process towards unknown or more subtle anomalies. Such a strategy is adopted in Chapter 4.

In cases when the scarcity and diversity of anomalies prevent obtaining a representative labelled data set, an alternative approach is a semi-supervised AD, also referred to as a particular case of One-Class Classification (OCC), Khan and Madden (2014). OCC algorithms aim to build models when copious examples of nominal behaviour or unlabelled data are relatively easy to collect while the anomalous class is either absent, poorly sampled or statistically not well defined. This unique situation constrains the learning objective to building the decision boundary around negative class such that it accepts as many objects as possible from this class while minimizing the chance of including any outliers. The model classifies data as either belonging inside the negative class decision boundary (inlier) or not (outlier). An advantage of the semi-supervised approach is that when once trained, it is not sensitive to the frequency of anomalies.

As observed by Tax (2001), the problems encountered in the conventional classification, such as the estimation of the classification error measuring the complexity of a solution, the curse

of dimensionality or the generalization of the classification method also appear in OCC and sometimes become even more prominent. The problem of learning is much harder than in binary classification because only one side of the boundary can be determined. As a consequence, it is difficult to decide how tight that boundary should fit the training data in each dimension. Furthermore, defining the negative region is difficult because some of the features may be irrelevant for discrimination, normal behaviour may not be static, or because the training data contains anomalies that are incorrectly modelled as normal behaviour. In fields where normal behaviour is continuously evolving (such as in monitoring), the model should be frequently modified. Finally, the training set must cover a broad spectrum of normal behaviour, which is often not a case either. Inappropriate training sets would always produce misleading results. In the context of monitoring the normal operation behaviour of the machine is easily obtainable. While most possible types of faults would not have occurred yet and waiting until such faults occur may involve high cost or risk to operation crew, the semi-supervised approach is a desirable extension of the supervised AD. Such strategies are explored in Chapter 4 and Chapter 5.

The unsupervised scenario uncovers anomalies in unlabelled test data. Unsupervised methods are based on two underlying implicit assumptions. Firstly the presumption that the negative instances are far more common than the positive ones. Secondly, the anticipation that there is a robust statistical variability between outliers and inliers. A typical unsupervised AD method is clustering, described in Section 3.3. An unsupervised approach reduces to finding sparse regions in large multidimensional data sets. In practice, the instances without an assigned cluster, or belonging to a small cluster, are declared as anomalies. However, it is difficult to determine the threshold between nominal data and the outliers. Frequently the performance suffers from high false alarm due to the abolishing of the underlying assumptions. As the noise represents the semantic boundary between normal data and anomalies, it is often modelled as a weak form of outliers. That does not always meet the strict criteria necessary for a data point to be considered anomalous enough. Besides, the inference time is sub-optimal on high dimensional data sets due to the computational complexity of clustering methods. Nevertheless, even though the unsupervised AD might not be the most robust technique, on some occasions is the only applicable one due to the lack of labels.

In the case of the CMS experiment, ML needs to cover known scenarios and extrapolate to new unseen problems. The supervised AD is a valid option, as the detector experts extensively studied specific anomalous scenarios. The CMS DQM framework keeps extensive archives of sub-detector specific quality-related quantities, e.g. the DT occupancy plots discussed in Chapter 4. Moreover, the imbalance between good and bad data may not be extreme, e.g. reaching 10% of anomalies for the DT system. These anomalies are then frequent enough for a sizable set of them to be used for supervised training. However, this setup has to be applied with caution.

The anomalous scenarios tend to be extremely disparate (property inherited from experiment complexity) and are rapidly evolving through time with new, unanticipated problems emerging regularly, as explained in Chapter 5. Also, the configuration of the LHC or the CMS experiment changes frequently implying a particular detector reaction. Consequently, different *good* behaviour is expected. Finally, some types of malfunctions occur rarely. All of the above suggests that the fully automated DQM is only promised by a smart mix of semi-supervised and supervised AD.

3.2 Performance Evaluation

Frequently the choice of the best model is a domain-specific task. In practice, this requires a good understanding of the data itself and types of deviations relevant to a target application. It is a subjective and heuristic process based on one's insight into the field. Assumptions are often imperfect, and the chosen pool of algorithms may model the underlying processes in a limited way. On the contrary, designing models with *a priori* knowledge is natural. Different algorithms may thus work better in some contexts than others. The stage of selecting the data model to be deployed in production is perhaps the most crucial one. Thankfully the ML field developed techniques to address the issue of choosing an optimal algorithm.

Whether a particular model is *suitable* depends on meeting the production requirements. Those usually include performance. The fundamental goal of learning is a generalization, i.e. being capable of inferring the knowledge learned from training data to unseen instances. A highly general model with many parameters will most likely overfit the data, as it will find a way to fit the outliers. A simpler model, constructed with an excellent intuitive understanding of the data, will lead to much better results. However, an underparameterized model, which fits the data poorly will declare standard patterns as anomalous. A typical empirical process is to evaluate the predictor on the test data (which was excluded from the training data set) of which the ground truth labels are known. The test error is a proxy of the generalization error. Crucially the test data should not overlap with the training data; otherwise, the estimated performance can be misleading.

That leads to another challenge. Establishing the generalization error of an AD algorithm is a difficult task, because outliers, by definition, are rare. Moreover, the boundary between normal and anomalous behaviour may be imprecise. The ground truth about outlying data points is often unavailable or obscure, which is especially true for unsupervised algorithms where the labels are not provided. It is often the case that no practical quantitative methods can be used to judge the effectiveness of the algorithms rigorously. Therefore, an intuitive and qualitative evaluation is the only possibility. An example of such an assessment is provided in Section 4.4. For supervised algorithms, the ground truth is available. Part of those labels can be used to perform the training, and the remaining can be used for evaluation. Even in unsupervised scenarios, the ground truth often becomes available after some time, some small fraction of example scenarios do exist or could be simulated (as in Chapter 6). Those rare labels may be used as surrogates for the ground truth anomalies. Subsequently, the natural question arises as to what is the correct protocol to evaluate the effectiveness of the algorithms.

The output of an AD algorithm can be one of two types: scores or labels. Scoring provides a continuous output, assigning a degree of *outlierness* to each sample. Thus the data set can be sorted and ranked according to the anomaly tendency. Scoring retains all the information provided by an algorithm but does not provide a concise summary of the samples which should be considered anomalous. After choosing an appropriate threshold value based on score distribution, the scores can be converted to binary labels. The labels indicate whether samples are normal (negative class) or anomalous (positive class). Binary labelling contains less information than a scoring mechanism, but it is the final result which is often needed in practical applications. Their outlier score and a varying anomaly threshold can be utilized to compare the performance of different algorithms. If this threshold is picked too restrictively

to minimize the number of declared outliers, then the algorithm will miss real outliers, False Negatives (FN), called type 2 errors. On the contrary, threshold highlighting most outlying points will likely lead to too many points declared as anomalous, False Positive (FP), also called type 1 error.

Typically, the performance comparison between different detection techniques is made using the Receiver Operating Characteristic (ROC) curve and its Area Under the Curve (AUC). The ROC curve is a graphical representation of the True Positive Rate (TPR), and False Positive Rate (FPR) values trade-off, by merely plotting the TPR as a function of the FPR. The TPR (also referred to as *sensitivity*, *recall* or *hit rate*) is the ratio between the number of items that are correctly assigned to the positive class, True Positive (TP), and the total number of items in the positive class:

$$TPR = \frac{TP}{P} \,. \tag{3.1}$$

The FPR (also referred to as *fall-out*) is the ratio between the number of items that are falsely assigned to the positive class, FP, and the total number of ground truth negative items. In the following chapters, the True Negative Rate (TNR) is used (also referred to as *specificity*) which measures the ratio of True Negative (TN) samples that are correctly identified as such:

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR \,. \tag{3.2}$$

The ROC curve clarifies the expense of having a higher number of correctly classified samples in the context of having more falsely predicted ones. After the anomaly score sorts the data points, the boundary between the classes is determined by a varying discrimination factor to calculate the corresponding rates and to produce the curve. The ideal algorithm would produce a line passing through the perfect classification point, which distinctly separates the positive from the negative class. In this context, the algorithm would produce a line such as one shown in Figure 3.2. An algorithm yielding random scores would produce the random guessing (the diagonal) line. If the target algorithm curve lies below that random guessing line, it performs oppositely to what it is expected. The lift obtained above the diagonal line provides a proxy of the accuracy of the approach. However, this lift depends on the complexity of the task. Hence, the algorithm should be compared to a set of baselines. The ROC AUC approximates the probability that an anomalous point would have a higher outlier score than an inlier. AUC is simply calculated as integral over the ROC line. The higher the AUC, the better the performance of the algorithm in general. However, in the context of stringent acceptance criterion, e.g. a maximum number of false alarms, the AUC may be misleading as the ROC curves may cross-over, i.e. the algorithms perform differently under different acceptance rules. For determining the optimal working point of the algorithm, one has to refer to cost-sensitive analysis, see Elkan (2001).

Alternatively, the efficiency of the algorithms can be measured in terms of precision, or Positive Predictive Value (PPV), and recall, and the Precision-Recall (PR) curve. With precision calculated as

$$PPV = \frac{TP}{TP + FP} \,. \tag{3.3}$$

The PR curve is simply a different way to characterize the trade-offs than the ROC curve, though the two can be derived from one another. The ROC curve has the advantage of being monotonic and more easily interpretable.

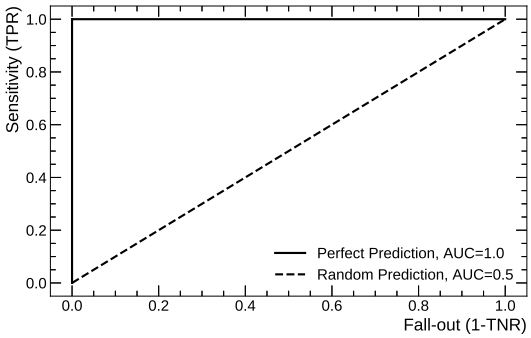


FIGURE 3.2: Examples of ROC curves with Sensitivity as a function of Fall-out. The two lines correspond to a perfect and a random prediction.

The interpretability of a detection model is extremely important in the context of AD. The fundamental goal of AD is to provide knowledge about alternative data generation processes to determine why a particular data point was highlighted as an outlier. That could provide further information for diagnosis and how to improve the overall production system. Different AD algorithms provide different levels of interpretability. Typically, models that work with the original attributes, and use fewer transforms on the data are better in this regard. While data transformations can sometimes enhance the discriminate power of detection algorithms, they often come at the expense of interpretability.

In the context of the CMS DQM, high performance is critical. It should be evaluated in terms of ROC AUC as the TP or FP requirements remain unspecified. Inference time is usually not an issue as in most cases. The alarm can be raised within the LS period (see Section 2.3). Finally, since this work revolves around monitoring, the interpretability of the algorithms should be considered.

3.3 Classical Anomaly Detection

In broad terms, each AD technique solves a specific formulation of the problem that accounts for different conditions. Those could include, e.g. nature of anomalies or outlier contamination in the training data set. Because of these constraints, the AD is usually not trivial, and many methods were proposed to target different scenarios, e.g. with specific latency constraints or availability of labels. The target application domain frequently determines these factors.

This section reviews the classical and well-established methods in the community: statistical, density, clustering, isolation and vector-based. It briefly overviews just the most popular algorithms from each category and identifies generic advantages and disadvantages of the methods. For a general survey, see Aggarwal (2016).

3.3.1 Statistical Methods

The early work on AD was done by the statisticians. In probabilistic and statistical models, the data are modelled in the form of a closed-form probability distribution. After Anscombe (1960), *an anomaly is an observation that is suspected of being partially or wholly irrelevant because*

it is not generated by the stochastic model assumed. The underlying principle of all statistical methods is that the inliers occur in high probability and anomalies in the low probability regions of a stochastic model. After fitting a model to the training data, the inference test determines if an unseen instance belongs to the distribution or not. Data points that report the low probability of being generated from the given model are declared as anomalies. This category divides methods into parametric and non-parametric techniques. Parametric techniques assume that the negative observations x are generated by a process that is modelled as a parametric distribution with parameters ϕ and the anomaly score of a test observation is the inverse of the Probability Density Function (PDF) $f(x, \phi)$. These methods make strong assumptions about the choice of the data distribution with which the modelling is performed, see Eskin (2000). The non-parametric statistical models work without the model's structure defined *a prioiri*. The distributions are instead determined from the training data. Such techniques typically make fewer assumptions, when compared to parametric techniques.

Frequently, the parametric techniques estimate the parameters of the model using Maximum Likelihood Estimation (MLE). Under the assumption that a Gaussian distribution adequately describes the data generation process, the typically used anomaly score is the distance from data point x to the estimated mean μ reported as a number of standard deviations σ. Since the $\mu \pm 3\sigma$ region covers approximately 99.7% of the normal distribution, the observations reporting distance bigger than 3σ are often regarded as anomalies. Alternatively, for univariate data sets, the **Grubb's Outlier Test** can be used. The anomaly score z (z-score), is computed as

$$z = \frac{|x - \mu|}{\sigma} , \tag{3.4}$$

and then tested according to t-distribution table reference. With data set size N and threshold t, each sample x for which

$$z > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2}{N-2+t^2}} , \tag{3.5}$$

is declared as anomalous. The threshold t is chosen from the t-distribution table, given the confidence level. When some prior knowledge about outliers is available, the **Dixon (Q) Outlier Test** can be used instead. The score q in the sorted data set of size m is computed as

$$q = \frac{x_n - x_{n-1}}{x_m - x_1} , \tag{3.6}$$

where x_n is the previously marked outlier, and x_{n-1} is the next candidate. Given m, a confidence level, and q-table reference, one can conclude if an observation is an outlier. In general Grubb's Test picks up extreme values earlier than the Dixon Test. For multivariate models, a commonly used metric is the **Mahalanobis distance**, see Laurikkala et al. (2000). The Mahalanobis distance d of an observation x and PDF P with sample covariance matrix S is defined as

$$d(x, P) = (x - \mu)^T S^{-1} (x - \mu) . \tag{3.7}$$

The distance grows as x moves away from the μ along each principal component axis. If axes are rescaled to unit variance, the Mahalanobis distance is equal to a Euclidean distance. Unfortunately, with increasing dimensionality of data, the Mahalanobis distance clusters

around a single value and the notion of outliers may become obscure.

The data can be modelled as a mixture of parametric statistical distributions as well, typically by the **Gaussian Mixture Model (GMM)** which characterizes the data as a generative process containing a mixture of M independent Gaussian clusters. Assuming that $\mathbb{N}(\mu, \sigma_{\mathbb{N}})$ is the distribution of normal data and $\mathbb{A}(\mu_{\mathbb{A}}, \sigma_{\mathbb{A}})$is the distribution of the anomalies, the distribution of the entire data set is

$$\mathbb{D} = \lambda\mathbb{A} + (1 - \lambda)\mathbb{N}\,, \tag{3.8}$$

where λ is the prior probability of the data point to be anomalous. The parameters of these distributions are learned with the **Expectation Maximisation (EM)** algorithm, Dellaert (2002). The membership probability to each of the clusters determines the anomaly score. EM algorithm alternates between performing expectation step by computing an estimation of likelihood using current model parameters and a maximization step by computing the maximum probability estimates of model parameters.

Generally, the non-parametric techniques do not assume the underlying distributions. A **histogram-based** technique is the most straightforward non-parametric statistical technique that uses histograms to maintain a profile of the normal data. Such techniques are also referred to as frequency-based. A basic histogram-based AD technique for univariate data consists of two steps. The first step involves building a histogram based on the different values taken by that feature in the training data. In the second step, the test instance is checked if it falls in any bin of the histogram. A variant of the basic histogram-based technique is assigning an anomaly score to each test instance based on the height (frequency) of the bin in which it falls. Thus, the size of the bin used when building the histogram is a critical setting. If the bins are small, many inliers will fall in empty or rare bins, resulting in a high false alarm rate. If the bins are large, many anomalous test instances will fall in regular bins, resulting in a high False Negative Rate (FNR).

Statistical methods are mathematically precise, well established and provide justifiable solutions, but they have their weaknesses. Simple assumptions about underlying data distributions are a massive advantage if they hold. However, in many real-world scenarios choosing the best statistic is often not a straightforward task even when the statistical assumptions are rational. In particular, constructing a hypothesis test for complex, high dimensional data sets is a challenge. Even more, when a model has to capture the interactions between different features. Anomaly score of statistical methods is associated with a confidence interval, which provides useful information. They can also be used in an unsupervised mode, avoiding the need for labels. Finally, regrettably, they offer poor algorithmic scalability, although the computational complexity depends on the chosen statistical model. In rare cases, when the model fits single parametric distribution from the exponential family, e.g. Gaussian or Poisson, it is linear in data size as well as a number of features. Complex distributions using EM, also have typically linear complexity, though they might be slow in converging depending on the criterion.

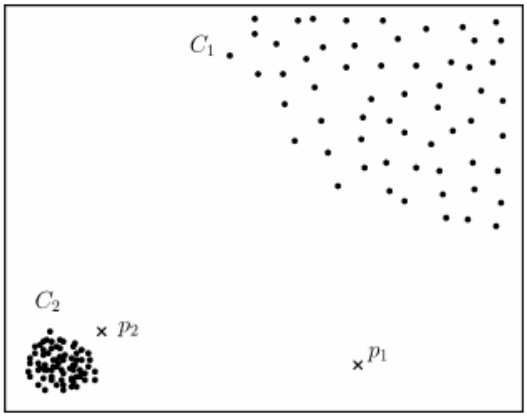


FIGURE 3.3: Two-dimensional data set with two different density clusters C_1 and C_2 and two outliers p_1 and p_2. From Chandola et al. (2009).

3.3.2 Density-Based Methods

The idea of a neighbourhood analysis is used in several AD techniques. The fundamental assumption here is that the outliers occur in sparse neighbourhoods and that they are distant from their closest neighbours while normal data instances appear in dense neighbourhoods. The neighbourhood is defined as the set of points lying near the object of interest. The techniques in this category require a distance or a similarity measure defined between two data instances. This distance (or similarity) can be computed in different ways. For continuous attributes, Euclidean distance is a popular choice, while the matching coefficient can be used for categorical attributes. Density-based AD techniques focuse on either using the distance of a data instance to its k-th nearest neighbor or computing a relative density of each data instance to compute its anomaly score.

K Nearest Neighbors (k-NN) is a global distance-based algorithm. k-NN is one of the modest and conventional non-parametric techniques for classifying samples. The anomaly score in the k-NN algorithm is equal to a distance to its k-th nearest neighbor, or the average distance to all of the nearest k neighbours, where k is a positive integer. For a given x, the distance to its k-th nearest neighbor is equivalent to the radius of a hyper-sphere, centred at x, and containing k other data points. This radius can be interpreted as an estimation of the inverse of the density of x in a global context. In the process of training k-NN classifiers, k is a critical parameter and varying this value can cause performance perturbation. The complexity of the unmodified method is $O(N^2)$, where N is the data set size. Thus variants of the technique were developed to improve efficiency, e.g. adding pruning Bay and Schwabacher (2003), partitioning Ramaswamy et al. (2000) or sampling Wu and Jermaine (2006) of the search space. As well illustrated in Chandola et al. (2009), k-NN performs poorly, in varying density clusters. In the two-dimensional data space in Figure 3.3, the instance p_2 is more likely to be assigned an inlier label than any point in the sparse cluster C_1.

A set of methods was proposed to overcome the problem described above. Most notably, the **Local Outlier Factor (LOF)** algorithm, proposed in Breunig et al. (2000). LOF is a density-based method that relies on the local nearest neighbours search. The general assumption for density-based methods is slightly modified here. Namely, the outliers are assumed to have a substantially lower density than their *local* neighbours. The anomaly score is equal to the ratio of average local density and the local density of the data instance itself. For

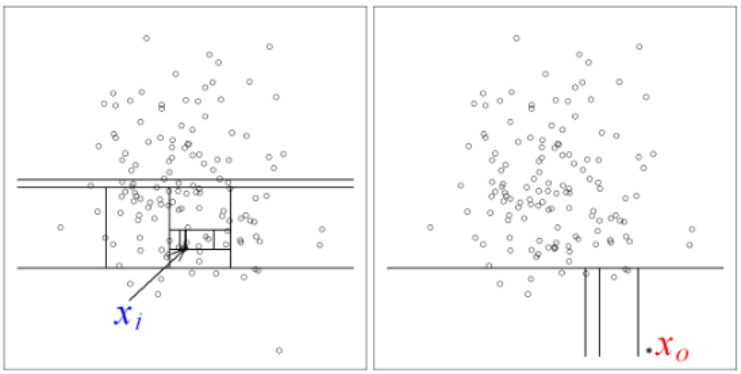
the radius of the smallest hyper-sphere centred at x, that contains k neighbours, the local density is calculated by dividing k by the volume of this defined hyper-sphere. Anomalies have lower local density than normal instances. Local density-based methods can detect outliers that were unseen by global methods, like k-NN. In the example from Figure 3.3, the algorithm will flag both p_1 and p_2 as anomalies. In need of additional interpretability, several extensions of LOF were developed. For instance, Local Correlation Integral (LOCI), which finds anomalous micro-clusters as well.

The essential advantage of the density-based methods is their unsupervised, data-driven approach and lack of assumptions about the generative distribution, or the statistical distribution of the data. These methods can explore outliers in their original spaces and demonstrated to work well on linearly separable distributions. When the anomalous regions are presented based on the original attributes, these techniques provide a high level of interpretability. Furthermore adapting them to a specific domain requires only redefining an appropriate distance measure. However, they tend to underperform with nonlinear structures, and the performance relies significantly on the distance measure chosen. Even if the Euclidean distance performs well, it is expensive to compute, especially in high dimensions with $O(N^2)$ computational complexity. Mitigating this issue via sampling or pruning can result in incorrect anomaly scores if the sample size is too small. In data sets with high noise to signal ratio, irrelevant attributes may mask the information contained in the relevant attributes. Besides, the latent correlation between the attributes results in an intrinsic dimensionality increase. Another weakness is the sensitivity of selecting the right value of k and the influence of duplicates on the performance. Finally, *lazy learning* (no explicit training process) may result in a high inference interval.

3.3.3 Clustering Based Methods

Clustering, see Jain and Dubes (1988), based methods bundle the data points into groups, according to a given similarity or distance measure (for a detailed presentation, see Goldstein and Uchida (2016)). The underlying assumption for performing AD with the techniques in this category is that similar data points tend to belong to similar clusters. As a consequence, the groups with normal instances are separable from anomalous instances. The AD is invoked only after establishing local centroids by an algorithm of choice. The anomaly score is determined by the distance from the tested data point and the closest centroid. Methods from this category, similarly to density-based methods, require distance computation between a pair of instances, and thus this choice is critical to method's performance. Thus the discussion from the previous section holds also here. The main difference between clustering and density-based methods is that the clustering methods segment the data points, whereas the density-based methods segment the space. Based on the choice of the clustering algorithm, the AD pipeline may differ.

In some cases outliers will not be assigned to any group; in other, they will be assigned to either a sparse or a small cluster. Assuming that the anomalies are uncommon, their clusters should be removed. The AD with clustering-based methods can be approached by training the model using unlabelled data (both inliers and outliers), or by using only normal data in a semi-supervised method.



(a) Isolating x_i (b) Isolating x_o

FIGURE 3.4: Two-dimensional, normally distributed data set with 135 points. An inlier x_i requires twelve random partitions to be isolated (a), while outlier x_o only four (b). Anomalies are more susceptible to isolation and hence are isolated faster. From Liu et al. (2012).

There is a long list of possible clustering methods among which the most conventional and straightforward is the ***k*-means** algorithm. It divides the data into k clusters based on a distance metric, typically Euclidean distance. After random initialization of k centroids, the data points are assigned to one of the clusters based on the distance to each centroid. Those centroids are then recalculated based on group membership. After several iterations, this process converges with a stable number of clusters and positions of the centroids. Again, the choice of the value of k is critical for the final performance. An example of using k-means for AD is given in Li (2010).

A key advantage of clustering-based techniques is their ability to operate in an unsupervised mode. Furthermore, clustering can be adapted to any complex data type. The very fast inference, which involves only computing distance to centroids, is a useful attribute in a production system. However, the computational complexity of these techniques is highly dependent on the effectiveness of the clustering algorithm and usually not small. The algorithms typically have quadratic complexity if the computation of pairwise distances for all data points is required, or linear with heuristic-based methods (such as k-means). Detecting anomalies as a byproduct of clustering is a disadvantage of itself as such algorithms are not optimized to do so. Unfortunately, the technique may be sensitive to outliers. Some methods will only be effective when the anomalies do not form clusters by themselves, especially when every instance is forced to be assigned to a cluster. As a result, a group of anomalies may be merged into a large cluster. If so, this will constitute a low TPR. This process is referred to as *masking*.

3.3.4 Isolation Forest

The assumption behind isolation based methods is that the anomalous instances are rare and very different from the remaining data points. Unlike the density and clustering-based categories, isolation does not rely on any distance or density measures. Instead, the methods fragment the data space to identify instances laying far from other data points. As illustrated in Figure 3.4, two instances, x_o and x_i require a different number of random partitioning iterations.

The **Isolation Forest (IF)** Liu et al. (2008, 2012), builds an ensemble of trees for a given data set. The anomaly score is represented as an average path from the root node to the terminating node on those trees. If the underlying assumptions hold, the outliers will have substantially shorter paths. Two parameters can be tuned to speed up the training procedure, i.e. the number of trees to build and subsampling size. The required evaluation parameter is the tree height limit. IF isolates a sample by randomly selecting a feature and randomly selecting a split value.

The IF offers linear computational complexity and a lack of assumptions about data distribution. Thus it scales up to handle considerable data size and high-dimensional problems. The method is unsupervised, brutally simple and since it is based on tree models offers high interpretability. However, the method measures the susceptibility of each data point to be isolated, and it will not work in some cases, i.e. conditional anomalies. The method is normalization sensitive and may underperform when not all the features contribute equally to anomaly scores. Finally, IF was designed to handle continuous-valued data only.

3.3.5 Support Vector Machine Based Methods

Support Vector Machine (SVM) was initially proposed by Vapnik (1995). SVMs first maps the input vector into a higher-dimensional feature space and then obtains the optimal separating hyper-plane in the high dimensional feature space. A decision boundary, i.e. the separating hyper-plane, is determined by support vectors rather than the whole training data set and thus is exceptionally robust to outliers. Originally an SVM classifier was designed for binary classification tasks, which makes it suitable for AD. The SVM also provides a user-specified parameter called a penalty factor, which allows users to make a trade-off between the number of misclassified samples and the width of a decision boundary. Kernels, such as Radial Basis Function (RBF), can be used to learn complex regions.

SVMs have been adopted to AD as OCC, i.e. learning a region that contains the training data instances and its boundaries. The basic technique determines if the test instance falls within the learned region. Tax and Duin (1999a,b) solve the OCC problem by constructing a hyper-sphere around the negative class data that contains almost all the data points with the minimum radius, known as Support Vector Data Description (SVDD). During inference, the model determines which side of that hyper-sphere a test instance lies and marks it as anomalous when it is outside the radius, as shown in Figure 3.5. An alternative approach, **One-Class Support Vector Machine (μ-SVM)**, was suggested by Schölkopf et al. (2001). The method constructs a hyper-plane instead of a hyper-sphere that separates the training data from the origin. Unlike SVDD, μ-SVM separates the regions that contain no data. Generally, SVMs may be sensitive to outliers and noise in training data sets. Strictly speaking, the μ-SVM is not an outlier detection, but a novelty detection method. The training instances are considered only negatively label and not contaminated by outliers. However, μ-SVM introduces a method to control the effect of outliers on the training procedure. $\nu \in (0, 1)$ is a parameter that controls a trade-off between maximizing the distance of the hyper-plane from the origin and the number of data points that are allowed to cross the hyper-plane (the FP). Finally, the μ-SVM has the valuable property of being a novelty detection algorithm: once trained, it is not sensitive to the frequency of anomalies.

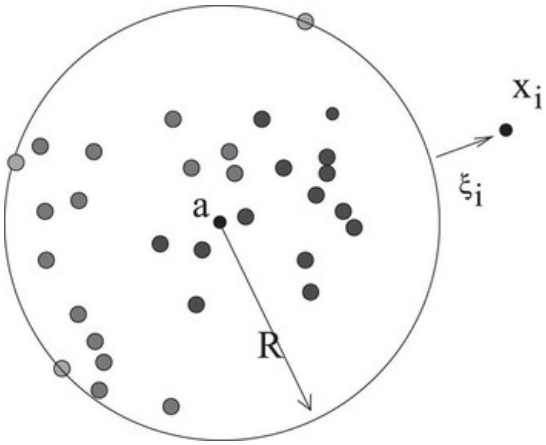


FIGURE 3.5: Illustration of SVDD on two-dimensional data set. The hyper-sphere contains the target data and is described by the centre a and radius R. One data point, an outlier x_i is outside the sphere and has $\xi_i > 0$, where ξ is a slack variable to account for errors. From Tax (2001); Khan and Madden (2014).

The hyper-sphere SVDD or hyper-plane μ-SVM estimate the support of the data distribution by a non-linear (kernel) transform of the data space. For instance, Tax (2001) considers Gaussian kernels working better than Polynomial ones, resulting in tighter descriptions, but requiring more data to support more flexible boundaries. Schölkopf et al. (2001) suggest the use of different kernels, corresponding to a variety of non-linear estimators. In this manner, the models can be made more flexible, depending on the nature of the input data.

3.4 New Approaches for Anomaly Detection

The critical challenge to overcome for HEP experiment monitoring AD is the vast volume of data. The proposed AD techniques need to be computationally efficient to handle these large-sized inputs. Indeed, excellent detection performance is necessary as well. However, the presence of noise in the data collected from the sensor makes AD more challenging. Finally, the simplicity of proposed solutions for implementation and debugging purposes and interpretability is required. The classical AD techniques discussed in Section 3.3 have a unique set of strengths and weaknesses. Unfortunately, they always miss at least one of the listed requirements. In most cases, it is the issue of high dimensionality.

The *curse of dimensionality* is often used as a vague indication that the high dimensional data cause problems in some situations. The term was first used by Bellman (1961) for the combinatorial estimation of multivariate functions. However, it is often used as a catch-all for multiple problems. Zimek et al. (2012) identified a number of them, which are summarized below.

- *The concentration of scores and distances* as derived values such as distances become numerically similar with low variance as the number of dimensions increase.
- A significant number of attributes may be irrelevant for the final sample classification and can mask relevant attributes (*noise attributes*).

- *Exponential search space* as the number of possible sub-spaces grows exponentially with the number of dimensions, and the search space can often be no longer be systematically scanned.
- For local methods, sub-spaces are based on neighbourhood-based methods (*definition of reference sets*).
- Particular objects occur more frequently in neighbor lists than others (*hubness*).
- Given the ample search space, for every desired outlier, a relevant hypothesis can be found (*data snooping bias*).
- It may be impossible to find a threshold between inliers and outliers due to low contrast (*thresholding*).
- The interpretability of scores is getting lost as the scores do not convey semantic meaning.

The unsupervised approaches based on neighbourhood, topological density estimation or clustering are not relevant for the problems at hand. These algorithms have quadratic complexity and poorly perform in high dimensions because of data sparsity. In high dimensions, all pairs of points become almost equidistant, see Aggarwal et al. (2001); Hinneburg et al. (2000). Moreover, a simple geometric distance in the feature space does not define a useful similarity metric. Furthermore, again because of data sparsity, even the definition of data locality may be ill-defined. Statistical techniques are valid only when the assumptions on the data hold. Unfortunately establishing those rules in the given context abolishes the simplicity principle. In situations when identifying a distance measure is computationally expensive, classification-based techniques may be a better choice. The computational complexity of classification-based techniques depends on the classification algorithm being used (for an overview see Kearns (1990)). While classification-based techniques have long training times, testing is usually fast. In such a case, the models can be trained offline, and testing in real-time is not an issue. In contrast, applying lazy-learning techniques, that do not have a training phase, becomes infeasible.

3.4.1 Deep Learning Anomaly Detection

AD is an especially non-trivial task in the presence of non-linear relationships in data. In the CMS DQM, the algorithms should remain sensitive to the local geometric relationship in the data related to the underlying apparatus. The algorithms must know how to obtain useful data representations from high dimensional input space. This representation learning view points this research towards DL framework, see Bengio et al. (2013). DL allows learning hierarchical discriminative features from data. That eliminates the need for manual feature engineering and allows them to learn from raw input data. DNN based AD offers state-of-the-art remedies to problems listed in the previous subsection. Neural networks, in the presence of enough training data, can cope with high data dimentionality. Besides, they offer simplicity and robustness of implementation and fast inference time. As shown in the experiments in subsequent Chapters (see 4.2.3 and 5.2.4) their peculiarities can be exploited, and a level of interpretability may be guaranteed. Finally, DL techniques can handle different types of data, e.g. images or numerical data, which is particularly important in the context of CMS DQM. For instance, the performance of the classical AD methods is sub-optimal when

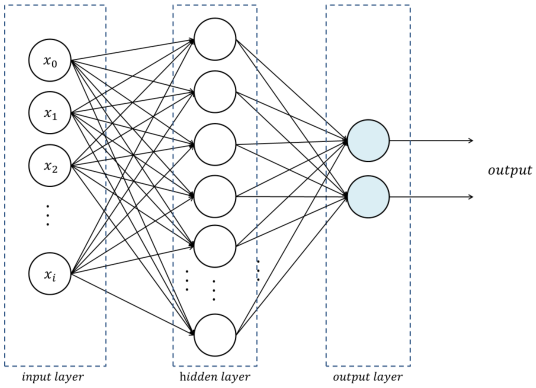


FIGURE 3.6: An illustration of the multi-layer feed-forward ANN from Sun (2019). Neurons are linked by weighted connections to form a network. In this example, there are no *in-* or *cross*-layer connections. An input layer receives input feature vector x, with each neuron corresponding to one feature. An output layer size corresponds to given ground truth y size. The layers between the input and output layers are called hidden layers. A non-linear transformation follows each output of a neuron in hidden, and output layers called an *activation function*, e.g. a sigmoid function.

the input is an image. DL can be particularly useful in cases when the boundary between nominal and anomalous behaviour is not precisely defined, and data is continually evolving (see Chapter 5). Finally, DNNs offer ways to cope with modelling complex apparatus, as discussed in Chapter 6. At this point, the relation of each approach to AD will be considered only from a general point of view. Chapters 4, 5 and 6 will match these general descriptions with the problem at hand in each case.

Neural networks also called Artificial Neural Networks (ANN), originated from simulating biological neural networks. The neural network function is determined by the model of a neuron, the network structure, and the learning algorithm. The neuron is also called a unit, which is the fundamental computational component in neural networks. Among the first attempts at designing a model with neuron-like structure, there is work of McCulloch and Pitts (1943). However, their model did not learn. Instead, neurons had a binary state (they fire or not) depending on the state of neighbouring neurons. Rosenblatt (1958) worked on a *perceptron* model, organizing neurons in two layers. Other notable works include Widrow and Hoff (1962), where a learning algorithm could tune weights. However, Minsky (1967) proved that simple perceptrons could not address the non-linearly separable problems. Only decades later, Rumelhart et al. (1988) developed a *backpropagation algorithm* which allowed multi-layer perceptrons to learn and so the non-linear separability curse was finally broken. Cybenko (1989) proved that multi-layer perceptrons with one hidden unit could be regarded as universal approximators of continuous functions. The term DL in the context of ANN was first introduced by Aizenberg et al. (2000), referring to architectures having multiple hidden layers (see Figure 3.6). This millennium brought big successes of DL, including great results on the *ImageNet* classification benchmark Krizhevsky et al. (2012) or mastering game of Go, Borowiec (2016). All of the above shows that the concept of DNN is well established in the ML community for decades. For a detailed historical survey, see Schmidhuber (2015).

The successes of DL are due in part of the simplicity of the backpropagation algorithm, which allows training neural networks efficiently. The goal of the neural network training procedure is to determine the values of the connection weights w and the biases b of the

neurons. The backpropagation computes the gradient of any loss function defined as a composition of differentiable functions. At first, the inputs are feed-forwarded from the input to the output layer, at which the error E is calculated by comparing the network output with the given ground-truth. Then, the error is backpropagated to the hidden layer and the input layer through Stochastic Gradient Descent (SGD), during which the connection weights w and biases b are adjusted to reduce the error on each layer ℓ. The process is accomplished by moving towards the direction with the gradient. The updating rule is specified as

$$\begin{aligned} w^{(\ell)}(e+1) &= w^{(\ell)}(e) - \alpha \frac{\partial E}{\partial w^{(\ell)}(e)} \\ b^{(\ell)}(e+1) &= b^{(\ell)}(e) - \alpha \frac{\partial E}{\partial b^{(\ell)}(e)} , \end{aligned} \tag{3.9}$$

where $\alpha(>0)$ is the learning rate of the algorithm. Such a process will be repeated in many rounds e (*epochs*) until the training process is terminated.

The choice of a DNN architecture primarily depends on the nature of input data and the application type, e.g. Recurrent Neural Networks (RNNs) for sequential or Convolutional Neural Networks (CNNs) for image data. For a general overview, see van Veen and Leijnen (2019).

CNNs are extending neural networks with convolutional and pooling layers. CNNs integrate the basic knowledge of merely the topological structure of the input dimensions and learn the optimal filters that minimize the objective error. They are thus very suitable for AD with images as inputs Kwon et al. (2018). Their ability to extract abstract hidden features from high dimensional input space enabled its use as feature extractors in AD context. Such an approach is shown in Section 4.4.

Deep architectures have become an increasingly popular method for AD tasks as they cope with the issues of classical methods. Generally, the DL based AD can be grouped into supervised, semi-supervised, hybrid, unsupervised and OCC categories, for a general survey see Chalapathy and Chawla (2019). Supervised AD involves training a neural network with a *softmax* activation function and binary (nominal and anomalous) or multi-class output layer. This approach is explored in Section 4.2. Semi-supervised, hybrid and unsupervised DL AD typically utilize properties of deep autoencoders. As the autoencoders are in particular interest of this thesis, they are carefully reviewed in Section 3.4.2. The hybrid methods use autoencoders as feature extractors. Those features are then attached as input to classical AD algorithms such as μ-SVM as demonstrated in, e.g. Andrews et al. (2016); Wu et al. (2015). In this context, the deep autoencoders have to ensure that the extracted representations are separable.

Furthermore, the computational complexity of a hybrid model includes the complexity of both the autoencoders as well as traditional algorithms used within. For this reason, those methods are not considered in this thesis. OCC based on neural networks, trains a DNN while optimizing a data-enclosing hyper-sphere Ruff et al. (2018) or hyper-plane Chalapathy et al. (2018) in the output space. The disadvantages of each of the group are inherited from general strategies, discussed in Section 3.1. Additionally, these approaches inherent a list of issues related to DL methods. That includes a non-trivial choice of architecture and hyper-parameters. Finally, while neural networks inference is very fast, they are characterized by long training times.

DL AD was reported particularly useful in the industrial context, e.g. Ramotsoela et al. (2018); Atha and Jahanshahi (2018).

3.4.2 Autoencoders and Anomaly Detection

Spectral AD aims to find the lower dimensional embedding of observable data x that separates anomalies from inliers. An example of such method is Principal Components Analysis (PCA), Pearson (1901). A *reconstruction* process brings the data from those embeddings back to the original shape of x. The output of the reconstruction is expected to be filtered, without noise. Reconstruction error, a distance between the original x and reconstructed data $\hat{x}$ is often used as an anomaly score to detect outliers. However, it uses only linear transformations which are often not sufficient. The alternative is the trade-off between interpretability and simplicity by learning an encoding using an autoencoder.

Autoencoders, Hinton (1990), are parametric maps from inputs to their representations, in the form of an ANN. Autoencoders are trained to perform an approximate identity mapping between their input and output layers. A deep autoencoder is composed of two neural networks: an encoder $\mathcal{E}$ that takes an input and maps it to a usually low-dimensional representation and a decoder $\mathcal{D}$ that tries to reconstruct the original input from the representation vector:

$$\hat{x} = \mathcal{D}(\mathcal{E}(x)) \text{ where } \hat{x} \sim x\,. \tag{3.10}$$

The model should prioritize which aspects of the input should be distilled to learn useful properties of the data, obtaining more abstract features in higher hidden layers leading to a better reconstruction of the data. A simple autoencoder with just a linear activation function in hidden units will mirror the behaviour of the PCA algorithm. While PCA is restricted to a linear dimensionality reduction, autoencoders enable both linear or non-linear transformations. However, it is often challenging to learn commonalities within data in a complex and high dimensional space and choosing network hyper-parameters for optimal results, such as the right degree of compression, is not trivial.

The reconstruction criterion is in itself not sufficient for learning useful representation, Bengio et al. (2013). To go beyond simple dimensionality reduction with *undercomplete* autoencoder while preventing over-fitting, various flavours of regularization are proposed (the literature being considerable, list contains most popular techniques), e.g. *sparse* Ranzato et al. (2006), *denoising* Vincent et al. (2010) or *contractive autoencoders* Rifai et al. (2011). Sparse autoencoders penalize the output of the hidden unit activations or the bias. Denoising autoencoders robustify the mapping by requiring it to be insensitive to small random perturbations. Contractive autoencoders pursue the same goal, by penalizing the sensitivity of learned features in a data-driven interpretation of the Tangent Propagation algorithm, Simard et al. (1998). Denoising and contractive autoencoders learn density models implicitly, through the estimation of statistics or a generative procedure, Alain and Bengio (2014). In practice the regularization changes the geometry of the loss gradient and allows the autoencoders to optimize their weights in different global minimum.

Although it has been argued that, even for pure neural networks, most of the training is devoted to learning a compressed representation, Tishby and Zaslavsky (2015); Shwartz-Ziv and Tishby (2017), autoencoders are particularly suitable for AD, Japkowicz et al. (1995);

Hawkins et al. (2002). When trained on the inliers, testing on unseen anomalous samples tends to yield sub-optimal representations and decoder outputs, that a different process likely generates a sample. Furthermore, the encoded representation space may distinguish the anomalous regions allowing for using hybrid methods. Although the autoencoders are an unsupervised method (or self-supervised), they are used in a semi-supervised AD strategy. In general, the underlying assumption is a high prevalence of normal instances over abnormal data in the training set. Not holding to this requirement would result in high FPR. Even a small number of anomalies contaminating the training data set can result in autoencoder learning to reconstruct anomalous observations accurately.

Autoencoders were reported useful in many AD problems, see Chalapathy and Chawla (2019). However, a more ambitious goal is to extract an explanatory representation of the anomalies with latent variables, in a probabilistic framework, i.e. VAEs where the learned representation is the posterior distribution of the latent variables given an observed input.

3.5 Variational Inference

VAEs allow for designing complex generative models and scaling them to large data sets. They allow for generating realistically looking synthetic images Gregor et al. (2015), fictional celebrity faces Hou et al. (2017) or music Roberts et al. (2017). This section presents the VAEs in detail as they are the main interest of Chapter 6. In most papers, the VAE is presented in isolation, but architecture is not a stand-alone discovery. The most natural presentation is followed here. VAEs are related to VI, although other presentations would be possible, e.g. in relation with autoencoders, Independent Component Analysis (ICA), or with an information theory vision. The goal is to isolate what is specific to the VAE, i.e. amortized inference, from the methods and techniques that have been developed for making VI first possible, then scalable. Thus, the first part of this section will describe VI in some details. The second part will survey the VAE, its extensions and the relation with the autoencoders. Finally, the VAE is discussed in the context of AD.

3.5.1 Approximate Variational Inference

Probabilistic graphical models express assumptions about the observed data and their hidden structure, in the form of a model. The corresponding posterior inference aims at inferring the hidden structure that best explains the observations. However, this posterior is generally not tractable and must be approximated. The two most prominent strategies in statistics and ML are Markov Chain Monte Carlo (MCMC) sampling and VI. MCMC is generally recognized as ill-suited to analyzing large data sets or complex models. Hence this section will focus on VI.

VI implements the general strategy of variational approaches, which is to recast the problem at hand into an optimization problem. For inference, the method needs to define a flexible family of distributions over the hidden variables, indexed by free parameters, Jordan et al. (1999); Wainwright et al. (2008). The problem boils down to finding the setting of the parameters (the member of the family) that is closest to the posterior, which is an optimization

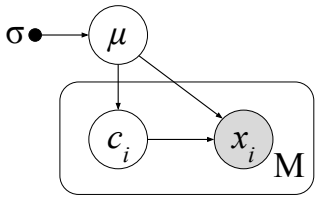


FIGURE 3.7: Graphical model for the Bayesian GMM. The distribution of each observation x_i depends on its corresponding local variable c_i and the global variables μ.

problem. It turns out that this optimization requires a preliminary approximation, on the optimization objective.

Bayesian Inference

The general scenario of Bayesian inference is as follows: observations $(x_1, \dots x_n)$ that are i.i.d. samples of a random variable x. The data are generated by a random process involving an unobserved multidimensional random variable z, which is called the *latent* variable. The overall probabilistic model is defined by

$$p(x, z) = p(x|z)p(z)\,, \tag{3.11}$$

where the likelihood $p(x|z)$ and the prior $p(z)$ are well-defined, i.e. parametric distributions. The inference problem is to compute the conditional density of the latent variables given the observed one, that is the posterior $p(z|x)$. Direct use of the Bayes rule

$$p(z|x) = \frac{p(x, z)}{p(x)} \tag{3.12}$$

is intractable because it requires the evidence $p(x)$. The intractability appears when the evidence is computed by marginalizing out the latent variables from the joint density

$$p(x) = \int p(x, z)\,\mathrm{d}z\,. \tag{3.13}$$

This evidence integral is generally unavailable in closed form or requires exponential time to compute.

An illustrative example of such is the Bayesian GMM, for unit-variance univariate Gaussian distributions. There are K mixture components; each parameterized by μ_k. The mean parameters are drawn independently from a standard Gaussian centred prior, with σ as an hyper-parameter. The generative process is as follows: to generate an observation x_i, first draw a cluster assignment c_i (here encoded as an indicator K-vector, all zeros except for a one in the position corresponding the cluster to which x_i belongs; then draw x_i from the corresponding Gaussian. Overall, the latent factors z are $\{\mu, c\}$, and the model is

$$\begin{aligned} \mu_k &\sim \mathcal{N}(0, \sigma^2) \\ c_i &\sim \text{Categorical}(1/K, \dots, 1/K) \\ x_i|c_i, \mu &\sim \mathcal{N}(c_i^T \boldsymbol{\mu}, 1) \end{aligned}$$

This example illustrates that the latent factors can be either global or point-wise: μ is shared by all the data points in the same cluster, while c_i is a point-wise parameter describing to which cluster the observation x_i belongs. The example illustrates the intractability, as well. The time complexity of numerically evaluating the K-dimensional integral is exponential, hence intractable. Figure 3.7 shows a graphical model of this setting.

Approximate Variational Inference: the ELBO

The inference problem is turned into an optimization problem by choosing a family $\mathcal{Q}$ of densities over the latent variables. A distribution q in $\mathcal{Q}$ is generally noted as $q(z|x)$ to highlight the fact that some or all the latent factors depend on the observations, even if no joint distribution is postulated. Free *variational parameters* parameterize these densities. The goal of optimization is to find the density $q*$, that is, a set of variational parameters, that is closest to true posterior $p(z|x)$. The most widely used measure of the discrepancy is the Kullback-Leibler (KL) divergence, so that:

$$q* = \text{argmin}_{q\in\mathcal{Q}}\mathbb{D}_{\text{KL}}(q(z|x)||p(z|x)). \tag{3.14}$$

The optimization complexity depends on the richness of the family.

However, the optimization problem is structurally intractable: the complete computation of the KL divergence is as intractable as the evidence. From the definition

$$\mathbb{D}_{\text{KL}}(q(z|x)||p(z|x)) = \mathbb{E}_q[\log q(z|x)] - \mathbb{E}_q[\log p(z|x)]\,, \tag{3.15}$$

and expanding $p(z|x)$ as $\frac{p(x,z)}{p(x)}$,

$$\mathbb{D}_{\text{KL}}(q(z|x)||p(z|x)) = \mathbb{E}_q[\log q(z|x)] - \mathbb{E}_q[\log p(z,x)] + \log p(x)\,, \tag{3.16}$$

where the evidence appears, it turns out that *minimizing* the KL divergence over $\mathcal{Q}$ does not require computing the evidence $p(x)$, as it is constant for q. Hence it is equivalent to maximizing the first part of the divergence:

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(x,z) - \mathbb{E}_q[\log q(z|x)]\,. \tag{3.17}$$

Intuitively, the first term rewards variational distributions that place high mass on configurations of the latent variables that also explain the observation. The second term rewards variational distributions that are entropic, i.e., that maximize uncertainty by spreading their mass on many configurations.

$\mathcal{L}$ is called the *Evidence Lower Bound (ELBO)*. Equation 3.16 can be rewritten as

$$\log p(x) = \mathbb{D}_{\text{KL}}(q(z|x)||p(z|x)) + \mathcal{L}\,. \tag{3.18}$$

As the KL divergence is always positive, the ELBO is a lower bound of the evidence.

Maximizing ELBO objective provides some intuitive interpretation as a classical tradeoff between likelihood and prior.

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_q[\log p(z)] + \mathbb{E}_q[\log p(x|z)] - \mathbb{E}_q[\log q(z|x)] \\ &= \mathbb{E}_q[\log p(x|z)] - \mathbb{D}_{\mathrm{KL}}(q(z|x)||p(z)) \,. \end{aligned} \tag{3.19}$$

The first term is an expected likelihood. As such, it favors densities on the latent variables that explain the observed data. The second term is the divergence between the variational density and the prior; as $\mathbb{D}_{\mathrm{KL}}$ is always positive, that is it brings q closer to the prior. The ELBO is a non-convex objective function and optimizing converges to a local optimum.

The Mean-field Approximation

To completely specify the ELBO as an optimization problem, the variational family (Q) family must be specified. A key idea behind variational inference is to choose the family flexible enough to capture a density close to the posterior, but simple enough for efficient optimization. Most work considers the *mean-field approximation*, where the latent variables are mutually independent, and its own set of parameters defines the distribution of each of them; generically, if z is m-dimensional, i.e. $z = (z^{(1)}, \dots z^{(m)})$,

$$q(z) = \prod_{l=1}^{m} q(z^{(l)}) \,. \tag{3.20}$$

For instance, for the mixture of Gaussians Blei et al. (2017) proposes the following family:

$$q(\mu, c) = \prod_{k=1}^{K} a_k(\mu_k) \prod_{i=1}^{n} b_i(c_i) \,, \tag{3.21}$$

where $a_k$is a normal distribution $\mathcal{N}(m_k, \sigma_k)$ over the k-th cluster mean μ_k and $b_i(c_i)$ is a distribution over the cluster assignment of the i-th observation defined by a K vector of assignment probabilities ψ_i. The variational parameters are composed of the set of $\{m_k, \sigma_k, \psi_i\}$.

In the simple optimization approach, the problem is wholly specified: maximize the ELBO for the parameter of the a and b distribution. Before sketching the corresponding algorithms, a natural question is: what are the properties that can be reasonably expected for its results? Besides empirical results, Wang and Blei (2019) (dubbed *Variational Bernstein–Von Mises* theorem) provide some theoretical foundation to this claim. The setting is frequentist, in the sense that the generating process has a true, fixed, and unknown value of the global parameters. Then, the variational posterior converges in total variation distance to the KL minimum of a normal distribution centred at the truth, and the estimator of the global parameters are consistent and asymptotically normal. For instance, for a full rank Gaussian variational family and a factorizable Gaussian variational family, then the Variational Bayes (VB) posterior for the factorizable family recovers the true mean and the marginal variance, but not the off-diagonal terms.

Optimizing the ELBO

Thanks to the mean-field hypothesis, a coordinate ascent algorithm can be used for this optimization. That is the strategy of the Coordinate Ascent Variation Inference (CAVI), discussed in Bishop (2006), which iteratively optimizes each factor of the variational density while keeping the others fixed, without any gradient computation. For instance, with the specific variational family of equation 3.21, the coordinate updates and the ELBO for given values of the parameters have closed forms. That is much more general than for this specific distribution and is valid for a large class of models and variational families (conditionally conjugate exponential), the coordinate updates and the ELBO for given values of the parameters have closed forms, Ghahramani and Beal (2001). The complete derivations and the CAVI algorithms are described in Blei et al. (2017).

The CAVI algorithm suffers from two limitations. First, it is not generic, requiring a complete specification of the model and variational family, restricted to the class mentioned above and to compute the corresponding closed forms analytically. Second, it is not scalable as each coordinate step iterates through the entire data set.

The alternative is a gradient-based optimization, which can scale with SGD. SGD maximizes a function using noisy estimates of its gradient, Robbins and Monro (1951) and is primary basic tool for scaling to large data sets, Bottou and LeCun (2003). In principle, to exploit SGD for optimizing ELBO with stochastic optimization, all that is needed is an unbiased estimator of its gradient w.r.t the variational parameters. Stochastic optimization of the ELBO has been initiated by the Stochastic Variational Inference (SVI) algorithm, Hoffman et al. (2013). For models with global and local parameters, SVI repeatedly subsamples the data to form noisy estimates of the natural gradient of the ELBO. The general flow of the algorithm is to get a (noisy) estimate of the ELBO by sampling a minibatch and optimize the associated local variational parameters, then compute the coordinate update for re-estimating the global variational parameters (see Figure 3.8).

A substantial body of research has been devoted to making VI more scalable and applicable to non-conjugate exponential family models. Recent developments target a more accurate VI, by relaxing the mean-field hypothesis, in particular with structured inference, and the use of other divergence metrics. An extensive review is available in Zhang et al. (2017). The discussion of structured inference is in Section 3.5.2 (in the restricted context of VAE). For a goal in this section, which is putting the VAE in the general context of VI, the most relevant work is related to making the VI more generic.

Toward Generic VI

Two approaches are presented, towards generic inference algorithm for which only the variational distribution has to be specified in closed form: Black Box Variational Inference (BBVI) and the reparameterization trick. The main idea is to represent the gradient as an expectation to estimate it with Monte Carlo (MC) techniques.

BBVI, Ranganath et al. (2014), proposes a method to obtain an estimator of the gradient-based on the *score function estimator* (also known as the REINFORCE or likelihood-ratio estimator), Fu (2006). It uses the identity $\nabla_\lambda F_\lambda(.) = F_\lambda(.)\nabla_\lambda \log F_\lambda(.)$, valid when F is a function differentiable in a parameter λ. The gradient of the ELBO from equation 3.17 can then be

written as an expectation:

$$\nabla_\lambda \mathcal{L}(q) = E_{q_\lambda}[\nabla_\lambda \log q_\lambda(z)(\log p(x,z) - \log q_\lambda(z))] \,. \tag{3.22}$$

The expectation can be then estimated by, e.g. naive MC (an M-sample average of the variational distribution): with $z_i \sim q_\lambda(z)$:

$$\nabla_\lambda \mathcal{L}(q) \approx \frac{1}{M} \sum_{i=1}^{M} \nabla_\lambda \log q_\lambda(z_i)(\log p(x,z_i) - \log q_\lambda(z_i)) \,. \tag{3.23}$$

It is resulting in an estimator that is applicable whenever $\log q_\lambda(x)$ is differentiable w.r.t λ.

Unlike CAVI or SVI, the analytical computations do not depend on the model, but only on the variational distribution. This basic version of the estimator suffers from high variance. Numerous various variance reduction techniques can be used to make the estimator more effective (see Maddison et al. (2017) for an extensive bibliography about the score function and dedicated variance reduction techniques). However, they have been largely superseded by the *reparameterization trick*, introduced in the context of variational inference independently by Kingma and Welling (2013); Rezende (2014); Titsias and Lázaro-Gredilla (2014).

The reparameterization trick allows to estimate the gradient of the ELBO also by MC samples but in a very different way. The trick states that in many cases a random variable z with distribution $q_\lambda(z)$ can be expressed as a deterministic function, parametrized by λ, of a random variable ϵ that comes from a noise distribution, i.e. $z = g_\lambda(\epsilon)$. Noise means that the distribution r of ϵ does not depend on any parameter. The condition is that the following identity holds:

$$E_{q_\lambda}[F(z)] = E_r[F(g_\phi(\epsilon)] \text{ for any function } F \,. \tag{3.24}$$

For example, if $z \sim \mathcal{N}(\mu, \sigma^2)$, then $z = \mu + \sigma\epsilon$, with $\epsilon \sim \mathcal{N}(0,1)$. Kingma and Welling (2013) provide a complete discussion of the variational families for which such a transformation exists.

The essential advantage is that the distribution of z does not depend on the variational parameter anymore, making it possible to reduce the problem of estimating the gradient of an expectation for the parameter of the distribution to the more straightforward problem of estimating the gradient for the parameter of a deterministic function. Applying identity from 3.24 to the ELBO expression from equation 3.17 results in

$$\nabla_\phi \mathcal{L} = E_r[\nabla_\phi(\log p(x, g_\phi(\epsilon) - \log q_\phi g_\phi(\epsilon))] \,, \tag{3.25}$$

which allows MC estimate, e.g. by averaging over samples of ϵ.

3.5.2 The Variational Autoencoder

The Variational Autoencoder (VAE) Kingma and Welling (2013); Rezende (2014) is an original construction that integrates the reparameterization trick, and amortized inference.

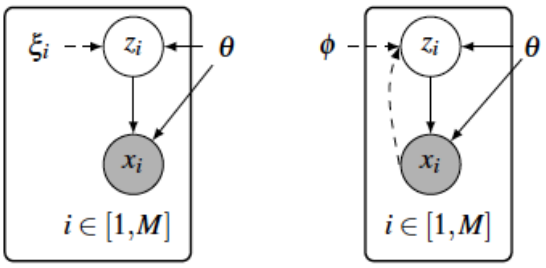


FIGURE 3.8: Graphical model for the classical VI setting (left) and the VAE (right). Each observation x_i depends on its latent variable z_i. Solid lines denote the generative model, parameterized by ϕ while dashed ones the variational approximations. From Zhang et al. (2017).

Amortized Inference

This approach is associated with a change in the generative model to an implicit point of view. Instead of looking at the evidence, primarily as the marginal of a closed-form distribution (equation 3.13), the data distribution is defined implicitly by

$$p(x) = p_\theta(x) = \int p_\theta(x|z)p(z)\mathrm{d}z \,. \tag{3.26}$$

This kind of parameter dependence in the factorization is sometimes termed *nonlinear factor analysis*, Gibson (1960); Krishnan et al. (2018). With this approach, the focus is on the expressiveness of the marginal $p_\theta(x)$. A neural network typically defines the likelihood $p_\theta(x|z)$, called the *generative network* (and the *decoder network* in the VAE context), which take the latent variables as an input and transform them in a non-linear way. In this setting, θ is the parameters of this network. With this model, the ELBO is

$$\mathcal{L}(x,\theta,\lambda) = E[\log p_\theta(x|z)] - \mathbb{D}_{\mathrm{KL}}(q_\lambda(z|x)||p(z)) \,, \tag{3.27}$$

where λ is the parameter of the distribution of the latent variables.

In classical VI, λ depend on the data points, as shown in Figure 3.8 (this dependency was exemplified in the expression of the posterior density 3.21 for the mixture of Gaussians case). As a consequence, the optimal parameter value λ^* is the result of direct pointwise optimization:

$$\lambda^* = \mathrm{argmax}\mathcal{L}(x,\theta,\lambda) \,. \tag{3.28}$$

One could formally write λ^* as a function of x (as $\lambda^*(x)$), but this would not have any operational sense, as its computation requires running the optimization algorithm for each data point x, which can be costly.

The amortized inference is, in some sense, a most radical approach. It switches from the transductive approach of VI, Gammerman et al. (1998), to an inductive one: it assumes that the latent variables can be approximately expressed (or say predicted) by a parameterized *function* of the data, that is making $\lambda(x)$ an actual function of x. The function is defined by a neural network, the *recognition* or *encoder* or *inference* network. In turn, this network is defined by its parameters ϕ. $q_\lambda(z|x)$, which should be noted as $q_{\lambda_\phi(x)}(z|x)$, will be further shortened as $q_\phi(z|x)$.

Algorithm 1 Minibatch version of training the VAE. From Kingma and Welling (2013).

$\theta, \phi \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M data points (drawn from full data set)
  $\epsilon \leftarrow$ Random samples from noise distribution $p(\epsilon)$
  $\mathbf{g} \leftarrow \nabla_{\theta,\phi}\mathcal{L}^M(\theta, \phi, \mathbf{X}^M, \epsilon)$ (Gradients of minibatch estimator)
  $\theta, \phi \leftarrow$ Update parameters using gradients $\mathbf{g}$
until convergence of parameters (θ, ϕ)
return θ, ϕ

Such a system is *trainable*. Once the λ_ϕ function is optimized, the latent variables can be predicted by passing new data points through the function, as shown in 3.8 right. An illustrative example is the so-called Deep Latent Gaussian Model (DGM), Rezende (2014). The generative model is defined for each data point x:

$$\text{where } p_\theta(x|z) \sim \mathcal{N}(\mu(z), \sigma^2(z)\mathbb{I}) \tag{3.29}$$

$$\text{and } p(z) \sim \mathcal{N}(0, \mathbb{I}) \,. \tag{3.30}$$

Despite its simplicity, the DGM model is very expressive, as an infinite mixture controlled by z.

Equation 3.29 shows that the likelihood depends on z through two functions $\mu(.)$ and $\sigma(.)$, which are the generative/decoder network. The data are then drawn from a normal distribution defined by the transformed latent variables $\mu(z)$ and $\sigma(z)$.

To approximate the posterior $p(z|x)$, the original VAE employs an amortized mean-field variational distribution:

$$q_\phi(z|x) = \prod_{i=1}^{N} q_\phi(z_i|x_i) \,. \tag{3.31}$$

The variational distribution is typically chosen as

$$q_\phi(z_i|x_i) \sim \mathcal{N}(\mu'(x_i), \sigma'^2(x_i)\mathbb{I}). \tag{3.32}$$

Similar to the generative model, the variational distribution employs non-linear mappings $\mu'(x_i)$ and $\sigma'(x_i)$ that are realized with the recognition network. The parameter ϕ summarizes the corresponding network parameters.

Training the VAE

During optimization, both the recognition network and the generative network are trained jointly to optimize the ELBO, principally by using the reparameterization trick, as sketched in Algorithm 1.

This basic algorithm applies SGD to an estimator of the ELBO obtained by applying the reparameterization trick. For instance, highlighting the dependence of the distributions on θ and ϕ, equation 3.19 can be rewritten as

$$\mathcal{L}(\theta, \phi) = E[\log p_\theta(x|z)] - \mathbb{D}_{\mathrm{KL}}(q_\phi((z|x)||p(z)) \,. \tag{3.33}$$

For each data point x, the ELBO can be estimated thanks to the reparameterization trick and MC estimation as

$$\widetilde{\mathcal{L}}(\theta,\phi) = -\mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)||p(z)) + \frac{1}{L}\sum_{k=1}^{K}\log p_\theta(x|z_k)] \tag{3.34}$$

$$\text{where } z = g_\phi(\epsilon_k, x) \tag{3.35}$$

$$\text{and } \epsilon_k \sim r(\epsilon) \tag{3.36}$$

In this last expression, it is assumed that the KL divergence can be computed analytically, thus is not part of the MC estimation. Then, classical minibatch gradient descent can be used, with the overall approximator of the ELBO over a full data set $X = (x_1, \dots x_N)$ being

$$\widetilde{\mathcal{L}}^M(\theta,\phi,X) = \frac{N}{M}\sum_{i=1}^{M}\widetilde{\mathcal{L}}^M(\theta,\phi,x_i)\,, \tag{3.37}$$

M running as the size of the minibatch.

Getting an analytical expression of the KL divergence requires a further assumption, for instance, that $q_\phi(z|x)$ is gaussian too. $q_\phi(z|x) \sim \mathcal{N}(\mu, \sigma^2)$ yields with standard computations:

$$-\mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)||p(z)) = \frac{1}{2}\sum_{i=1}^{I} 1 + \ln\sigma_i^2 - \mu_i^2 - \sigma_i^2\,. \tag{3.38}$$

Autoencoders and VAEs

The overall architecture and the joint training of the generative and recognition network are very similar to the autoencoders. Unregularized autoencoders minimize only the reconstruction cost. That results in different training points being encoded into non-overlapping zones chaotically scattered all across the latent space with holes in between where the decoder mapping has never been trained. The various regularizations target this issue to force the networks to learn a useful and compact representation of the data. VAE are probabilistic models that learn a representation of the distributions involved in the implicit model of equation 3.26. VAEs include the idea of noisy autoencoders Vincent et al. (2010), by injecting noise into this intermediate layer, which has a regularizing effect. However, the probabilistic orientation is due to the inclusion of the KL divergence term between the prior and the approximate posterior, which forces a homogeneous distribution in latent space that generalizes better to unseen data. In the direct context of the autoencoders, the KL divergence acts as a regularizer, while the second term in equation 3.17 is an expected negative reconstruction error.

Beyond the Vanilla VAE

As described in Cremer et al. (2018); Krishnan et al. (2018), the VAE has two source of errors: the approximation gap, which comes from the inability of the approximate distribution family to exactly match the true posterior, and the amortization gap, which refers to the difference caused by not optimizing for each data point independently, and consequently

not finding the optimal member of this family. Despite some theoretical results, Cherief-Abdellatif (2019), showing that the ELBO maximization strategy is robust to model misspecification, both sources of errors have been addressed through more expressive variational distributions, more expressive models, or extended optimization algorithms.

VAEs are limited by strong factorization. It cannot capture dependencies between latent variables. In general, using more flexible variational distributions reduces not only the approximation error but also the amortization error. More specifically, a well-known issue is the frequently observed problem of under-estimation of the variance of the posterior distribution, Turner and Sahani (2011)

In the strict VI framework, there have been attempts to fit more expressive approximating distributions which capture correlations such as matrix-variate Gaussian distributions, Louizos and Welling (2016). Fitting those models can be expensive without further approximations. Zhang et al. (2018) propose a method based on natural gradient and the Kronecker-factored approximation to the Fisher matrix (to perform efficient approximate natural gradient updates). Implicit distributions can be used in VI since a closed-form density function is not a strict requirement for the inference model; all that is needed is to be able to sample from it and to compute the gradients. However, because the divergence becomes intractable, this results in a different training procedure. For instance, Adversarial Variational Bayes (AVB), Mescheder et al. (2017), introduces an auxiliary discriminative network that rephrases the maximum-likelihood-problem as a discriminator T in the spirit of Generative Adversarial Networks (GAN), Goodfellow et al. (2014), to discriminate the prior from the variational distribution.

Normalizing flow from Rezende and Mohamed (2015); Kingma et al. (2016) presents another way to utilize flexible variational distributions. A normalizing flow transforms a probability density through a sequence of invertible mappings. There exist transformation function f such that the generated function is a multimodal distribution of arbitrary complexity, but that the Jacobian involved in the change of variable remains easily computable. Moreover, the variational density can be estimated due to an invertible transformation function, and expectations can be computed efficiently.

Importance Weighted Variational Autoencoder (IWAE) was initially proposed to tighten the variational bound Burda et al. (2016). IWAE has the same architecture as the VAE, but with a strictly tighter log-likelihood lower bound. The recognition network uses multiple samples to approximate the posterior. Specifically, IWAEs require L samples from the approximate posterior, which are weighted by the ratio

$$\hat{r_l} = \frac{w_l}{\sum_{l=1}^{L} w_l} \text{ where } w_l = \frac{p_\theta(x, z_l)}{q_\phi(z_l|x)} \,. \tag{3.39}$$

With this reweighting, the log-likelihood is approached in the infinite limit on L. Cremer et al. (2017) shows that IWAE can be reinterpreted to sample from a more flexible distribution which converges pointwise to the true posterior.

On the other hand, it has been argued that factorization is a desirable property, as it favors disentangling: a representation that is adequate to latent factors that are assumed to be independent. A considerable literature has been devoted to this goal. Disentangling 3.5.2 and the alternative CVAE is discussed in Section 6.2.

A more profound problem is the model choices, which include the prior $p(z)$ and the conditional likelihood $p(x|z)$. In both cases, the goal is to devise general techniques that allow integrating some knowledge about the structure of the distributions that can be expected from a specific application. For instance, Johnson et al. (2016) consider video and speech. In both cases, there is a natural hierarchy of information: in video, the complementary tasks are learning the image manifold (low level) and a structured dynamics model (high level), In speech, the high-dimensional data lie near a low-dimensional manifold, but the discrete latent dynamical structure (phonemes, words, and grammar) should also be considered. Technically, Johnson et al. (2016) utilize a structured prior for VAEs, combining the advantages of traditional graphical models and inference networks. These hybrid models overcome the intractability of non-conjugate priors and likelihoods by learning variational parameters of conjugate distributions with a recognition model. An analogous approach is sketched in Butepage et al. (2018).

Other approaches question the assumption that the conditional likelihood factorizes over dimensions. Intuitively, this is a drawback when a structured output model is closer to the data, e.g. for images, where there are dependencies between neighboring pixels, or for language modelling. The general strategy is to couple advances in deep architectures, such as autoregressive networks Gregor et al. (2014) and predominately recurrent networks when the application is amenable to sequence modelling, e.g. for speech and writing Bowman et al. (2016); Chung et al. (2015); Fabius et al. (2015); Gregor et al. (2015); Xu and Tan (2019), but also for images with the PixelVAE Gulrajani et al. (2017). In practice, this means that the decoding distribution is an RNN with autoregressive dependency, i.e. $p(x|z) = \prod_k p_\theta(x_k|x_{<k})$, where k runs over the dimensions of observation x.

The expressiveness of the model can have unintended consequences by weakening the balance between the inference and recognition network. In summary, if the decoder is powerful, as exemplified just before, the inference network can fail to learn an informative posterior $q_\phi(z|x)$ and simply fall back to the prior $p(z)$. The problem has bee detailed early Bowman et al. (2016), and noticed in many other cases, e.g. Gulrajani et al. (2017): in most cases when an RNN autoregressive decoding distribution is used, the latent code z is completely ignored, and the model regresses to be a standard unconditional RNN. In Bowman et al. (2016), this has been attributed to optimization issues (the dying unit problem, see below). More recent work Chen et al. (2017) gives an information-oriented interpretation showing that the problem is more fundamental: the latent code should still be ignored at optimum for most practical instances of VAE that have intractable true posterior distributions and sufficiently robust decoders. Techniques to cope with this issue will be discussed in Section 6.2.

A separate issue is the dying units problem, which is related to the practical limits of the optimization process. Even if the decoder is not robust, in the early stages of the optimization where the approximate posterior does not yet carry relevant information about the data, units are regularized towards the prior. They might not be reactivated in the later stages of the optimization, as no gradient signal passes between the decoder and the encoder. The overall system is trapped in a local and inadequate minimum. Bowman et al. (2016) apply a classical annealing scheme to escape the minimum, by solely weighting the KL term and increasing the weight at a constant rate (a hyper-parameter). Thus, in the beginning, the model behaves as a simple autoencoder; increasing the weight forcing the model to smooth out its encodings towards the prior. Although ad-hoc, this method has the advantage to

mostly follow the VAE architecture, instead of proposals for alternative architectures such as Sønderby et al. (2016); Krishnan et al. (2018); Mansbridge et al. (2019). In particular, Krishnan et al. (2018) propose a warm-start of the generative network optimization: at each step of the joint optimization, the generative network is initialized by the output of an SVI run on the considered instance.

Disentanglement Objective

From a representation learning point of view, Bengio et al. (2013), the VAEs should produce a representation that is both *disentangled* and *interpretable*. The representation should contain all the information present in data x in a compact and interpretable structure while being independent of the task for which the representation is ultimately used. Such a representation would be useful for a variety of downstream tasks, such as transfer and few-shot learning or dealing with nuisance parameters. Bengio et al. (2013) and Locatello et al. (2019) present an extensive bibliography concerning these motivations in the general context of representation learning.

A large body of research has been devoted to disentanglement and interpretability. Recently, theoretical arguments and empirical evidence presented by Locatello et al. (2019) indicate that a fully general disentangling method is virtually inaccessible. However, in principle, it is possible to eliminate the obstacle using a priori knowledge and partial supervision that can be embodied in structured models, which are represented in the given context by the CVAE, described in Sohn et al. (2015). A recent result in the context of ICA suggests that the approach has theoretical foundations, see Khemakhem et al. (2019).

Given the observable data are generated from independent factors of variation, a disentangled representation should promote to separate these distinct factors in different independent variables in the representation. A change in a single underlying, unobservable, factor of variation should lead to a change in a single variable in the learned representation. For example, a model trained on character images might learn independent latent units sensitive to separate data generative factors, such as character label, width and angle. A disentangled representation is therefore necessarily factorized, and hopefully interpretable as independent ground truth generative factors.

VAE disentanglement has been addressed in numerous recent contributions, see Tschannen et al. (2018); Locatello et al. (2019). The various motivations lead in all methods to regularize the approximate or aggregate posterior distribution in the VAE objective function. The regularization addresses the KL divergence term as the components of the ELBO may become unbalanced, and the KL term tends to vanish (see Section 3.5.2).

The β-VAE proposed by Higgins et al. (2017) weights the KL divergence with a constant hyper-parameter $\beta > 1$ to put pressure on the posterior by enforcing the closeness with the factorized Gaussian prior $p(z)$:

$$\mathcal{L}_{\beta\text{-VAE}}(x, \theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \beta \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)||p(z)) \,, \tag{3.40}$$

In Burgess et al. (2018), it has been extended with progressive extension of the encoding capacity so that the encoder can focus on learning one factor of variation at the time.

However, as described in Kim and Mnih (2018), penalizing the KL term can have counterproductive consequence. The expectation of the KL term can be broken down as derived by Hoffman and Johnson (2016):

$$\mathbb{E}_{p_(x)}[\mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)||p(z))] = I_{q_\phi}(x;z) + \mathbb{D}_{\mathrm{KL}}(q_\phi(z)||p(z)) \,, \tag{3.41}$$

where $p(x)$ is the empirical probability and $I_{q_\phi}(x;z)$ is the mutual information under the joint distribution $p(x)q(z|x)$. Penalizing the second term is desirable, as it pushes q towards the factorial prior, but penalizing $I_{q_\phi}(x;z)$ is harmful, as it reduces the coding effectiveness of z. A common theme behind approaches addressing this effect is to enforce a factorized *aggregated posterior* $q(z) = \mathbb{E}_x[q(z|x)]$, at the cost of supplementary intractability and accordingly approximations. The FactorVAE, described in Kim and Mnih (2018), adds an extra penalization of the Total Correlation (TC) Watanabe (1960) of the latent variables (z_k), that is $\mathbb{D}_{\mathrm{KL}}(q(z)||\bar{q}(z))$, where $\bar{q}(z) = \prod_k q(z_k))$. As the objective is intractable, Kim and Mnih (2018) approximate the density ratio that arises in the KL term with adversarial training. The discriminator is trained to classify between samples from $q(z)$ and $\bar{q}(z)$, thus learning to approximate the density ratio needed for estimating TC. A similar analysis of Chen et al. (2018) proposes just a different approximation technique. The DIP-VAE Kumar et al. (2018) directly penalizes the mismatch between the aggregated posterior and the factorized prior, at the cost of two additional hyper-parameters. Finally, the InfoVAE Zhao et al. (2017) and the Wasserstein autoencoder Rubenstein et al. (2018), although with very different initial motivations, simply drop the mutual information term in the VAE objective.

There is no formal definition of a performance metric, and different approaches were proposed to measure disentanglement. As observed by Locatello et al. (2019), they appear to be correlated, but the level of correlation changes between different data sets. For instance the β-VAE measures disentanglement as the accuracy of a linear classifier that predicts the index of a fixed factor of variation. Kim and Mnih (2018) use a majority vote classifier on a different feature vector. Chen et al. (2018) measures the normalized gap in mutual information between the highest and second-highest coordinate in $r(x)$ for each factor of variation.

The recent results of Locatello et al. (2019) show that disentanglement learning is both fundamentally impossible for arbitrary generative models and not achieved in practice. From the fundamental point of view, the main result (Theorem 1 in Locatello et al. (2019)) essentially states that, for any fully disentangled model, there is an equivalent generative model with the latent variable $\hat{z} = f(z)$ where $\hat{z}$ is completely entangled to z: a change in a single dimension of z implies that all dimensions of $\hat{z}$ change. That is reminiscent of the non-identifiability in non-linear ICA Hyvärinen and Pajunen (1999).

From the practical side, Locatello et al. (2019) conducted an extensive experiment on the various disentangling-oriented algorithms based on regularization. As described in the example of Section 3.5.2, the actual representation used for downstream tasks, and the coupled training of the encoder and decoder networks, is the mean vector of the Gaussian encoder. The question is thus if this mean representation is decorrelated. Locatello et al. (2019) show that the total correlation of the mean representation generally increases with the regularization strength, and is higher than the one of the vanilla VAE. The only exception being the DIP-VAE, which directly enforces disentanglement. That is also true if the total correlation is replaced by the average mutual information between different dimensions of the representation.

3.5.3 Variational Autoencoders and Anomaly Detection

The dissemination of the generative models, and specifically the VAE, offer a more broad and principled avenue to autoencoding-based AD. A straightforward approach for VAE based AD is described in An and Cho (2015). It considers a simple VAE, and the MC estimate of the expected reconstruction error (termed reconstruction probability). Experiments on MNIST and KDD demonstrate a majority of the superior performance of VAEs over autoencoders and spectral methods.

However, Wang et al. (2019) argue that the probabilistic generative approach of VAE could suffer from intrinsic limitations when the goal is AD, with two arguments. Firstly, because the model is trained only on inliers, the representation will not be discriminative, and will essentially overfit the normal distribution. Secondly, the representation might even be useless, falling back to the prior; technically because the generator is too powerful, the regularization by the $\mathbb{D}_{\mathrm{KL}}$ vanishes Zhao et al. (2017).

Kawachi et al. (2018) address this issue with specific hypotheses on the distributions of inliers and anomalies. A more general approach Hendrycks et al. (2018); Wang et al. (2019) are to learn a more conservative representation by exposing the model to out-of-distribution (abnormal) examples, still without knowledge of the actual anomaly distribution, with adversarial architectures and specific regularizations. While Hendrycks et al. (2018) simply define an ad-hoc regularization and hyperparameter optimization, Wang et al. (2019) propose an adversarial architecture for generating the anomalies and exploiting them to create a less overfitted representation. Neither of these approaches would meet the robustness and simplicity specifications of the motivating application.

A body of work, e.g. Nalisnick et al. (2019); Snoek et al. (2019), discusses the general ability of the deep generative model to select Out of Distribution (OOD) data using the model likelihood. In Nalisnick et al. (2018); Choi et al. (2018); Shafaei et al. (2018), the authors showed that the generative models are not able to give useful estimates of the predictive uncertainty. Their tests showed that state-of-the-art models trained on one data set assign higher likelihood to samples from different data sets. For instance, a model trained on a data set containing images of objects and animals assigns a higher likelihood for images containing house numbers coming from a different data set.As a consequence the VAEs ability to perform safe AD is an open research question.

CHAPTER 4

Detector Components Anomaly Detection with Convolutional Neural Networks

Most of CMS DQM failure detection algorithms focus on the interpretation of detector data organized in the form of multidimensional histograms. The DQM visualization tool, described in De Guio (2014), displays those histograms organizing them in views dedicated to various aspects of the detector. Very often, the data are structured to provide the experts with an overview of the overall detector allowing to monitor its performance in different regions. For example, histograms can display two-dimensional maps with information about the performance of read-out channels depending on their geographical position. These kinds of histograms, displaying data as a function of the detector topology, can be regarded as images and the AD performed by the expert is very often related to identifying and discriminating healthy patterns from problematic ones. Detector experts input their knowledge of the detector into binary classification algorithms targeting common and foreseen failure scenarios.

This chapter will focus on applying image like detection techniques on geographically organized histograms filled with real collision data. The DL techniques are used for supervised and semi-supervised identification of problems in online detector monitoring (see Section 2.4.2). The attention settles around current advances in image classification (for a general survey see LeCun et al. (2015)) and the technical knowledge about the geographical organization of the detector.

A class of these problems is based on counting the number of electronic hits per read-out channel. This problem is common to all sub-detector technology and is referred to as *occupancy monitoring* in CMS jargon. Data recorded by the CMS DT chambers of the muon spectrometer during the data taking campaign of the LHC Run II are used as a case study. Although the experimental demonstration of the results presented in this chapter is tied to the peculiarities of the DT sub-detector (see Section 2.3.2), the procedure that is discussed here has a broader application scope. That is because the typical issues encountered in occupancy monitoring are analogous. This assumption was successfully tested for other sub-detector of the CMS experiment.

The goal is to improve detection specificity and sensitivity and propose an algorithm independent from a stringent assumption on the nature of the anomalies. In summary, the main aspects of this work are:

- exploiting the geographical information of the detector assessing the (*mis*) behaviour with high-granularity and then combining the results to probe different detector components;
- detecting various types of anomalies, affecting the detector at different scales (ranging from a few read-out channels to collective behaviours of a significant portion of the system) by combining different algorithms;
- justifying that image-like processing achieves considerably better performance compared to the current threshold-based statistical test (later called the *legacy strategy*) and allows to tune the working point in terms of specificity (depending on the deployment strategy); *and*
- introducing a novel, modular algorithm to the CMS DQM toolbox.

To this purpose, three complementary approaches are considered.

- The *local* approach considers the AD problem at the highest reasonable granularity. The input dimension (i.e. the number of features) is low, allowing for comparison between various algorithms, including the ones sensitive to the number of features. A well-known list of problems is targeted using data labelled by detector experts. This approach is explored in Section 4.2.
- The *regional* approach extends the local approach to account for problems seen when a broader view of the sub-detector is available. Simultaneously the labelling problem is overcome using semi-supervised AD. The approach is described in Section 4.3.
- Finally, the *global* approach further explores methods used for the regional approach and simultaneously use all the information coming from all the read-out channels in the sub-detector. Visual identification is used for emerging and novel problems along with the knowledge about the organization of the CMS detector. This approach is outlined in Section 4.4.

4.1 Data Set and Preprocessing

The DT system is an example of compound hardware that challenges scientists with the arduous work of establishing monitoring rules. For each chamber k in a given run the current DQM infrastructure Rovere (2015) records an occupancy matrix C^k, which contains the total number of particle hits at each channel for a given LS (arbitrary duration of time, see Section 2.4) or set of consecutive LSs. The occupancy matrix can be viewed as a varying size two-dimensional array organized with a layer (row) and channel (column) indices:

$$C^k = \{x^k_{i,j}; 1 \leq i \leq l, 0 \leq j < n_i\}\,, \tag{4.1}$$

where $l = 12$ is the number of layers and n_i is the number of channels in layer i. In general, chambers and their components are labelled as C^k and $x^k_{i,j}$. For simplicity, the k index is omitted when discussing problems related to individual chambers. Figure 4.1 shows examples

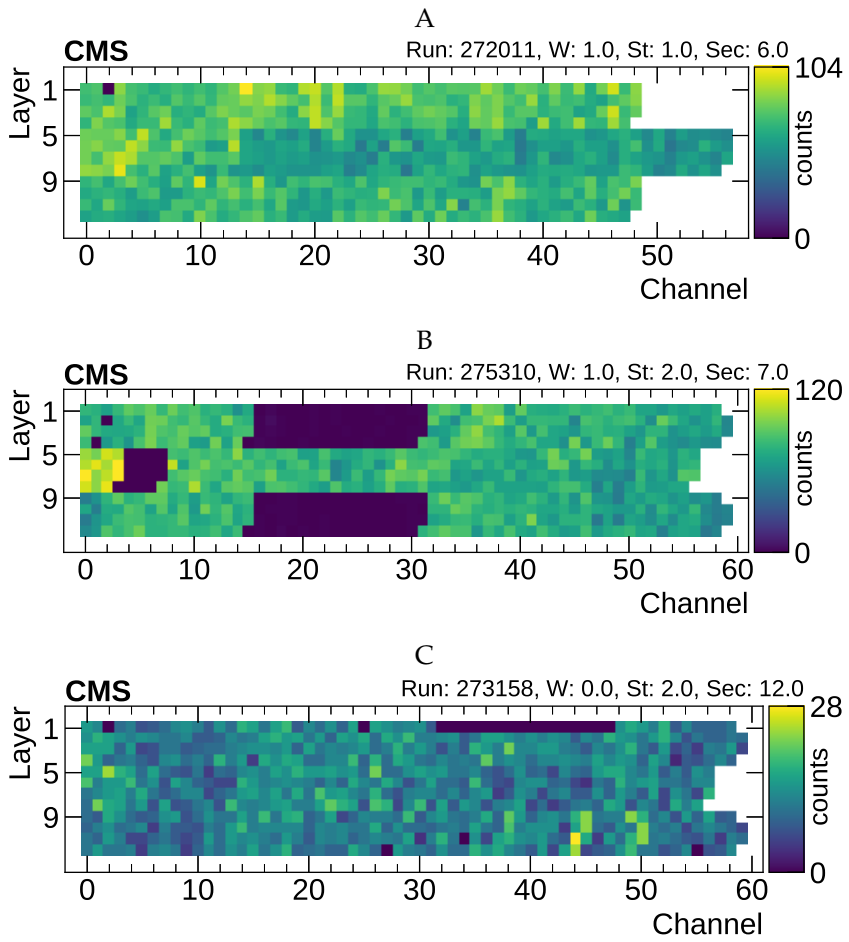


FIGURE 4.1: Example of visualization of input data for three DT chambers. The data in (A) manifest the expected behaviour despite having a dead channel in layer 1. The chamber in the plot in (B) instead shows regions of low occupancy across the 12 layers and should be classified as faulty. According to the run log, this effect was induced by a transient problem with the detector electronics. (C) suffers from a region in layer 1 with lower efficiency, which should be identified as anomalous.

of occupancy matrices, represented as two-dimensional occupancy plots. The utilized data set consists of 21000 occupancy matrices for the 250 chambers across 84 runs.

Histogram-based representation of detector data belonging to different subcomponents is often not wholly homogeneous in size (e.g. different DT chambers can have a different number of cells per layer) and can display data integrated over varying periods.

Preprocessing of data sets is a common requirement for ML algorithms, and in this case the preprocessing will have to deal with these irregularities in the feature space and additionally, depending on the algorithms, apply denoising, e.g. forests based algorithms benefit from it. Furthermore, some algorithms require features to be on a similar scale (e.g. SVM or neural networks) if they exploit distances or similarities between data samples. DT data set is no different. For this propose, a preprocessing procedure is used for the three approaches (for visual interpretation, see Figure 4.2).

- *Standardization of the chamber data*: the number of channels x in a layer varies not only layer-by-layer within the chamber but also depends on the chamber position in the

detector. This quantity falls between 47 and 96. Fixed-input dimensionality is enforced by applying a layer-by-layer one-dimensional linear interpolation to match the size n_s of the shortest layer in the data set. The smallest layer is chosen to simplify the models later in this study. After starting from the recorded matrix x, a standardized matrix $\tilde{x}$ is defined as

$$\tilde{x}_{i,l} = (\alpha - \lfloor \alpha \rfloor)(x_{i,\lfloor \alpha \rfloor} - x_{i,\lceil \alpha \rceil}) + x_{i,\lfloor \alpha \rfloor} \, , \tag{4.2}$$

where α is an interpolation point, defined by $\alpha = a\frac{n_i}{n_s}$. It was verified that this method does not compromise sensitivity to tiny problematic regions despite a small reduction in the amplitude of the anomalies.

- *Smoothing*: according to CMS DT experts, misbehaving channels are problematic only when a spatially contiguous cluster of them is observed. Instead, isolated misbehaving channels are not considered a problem. The reason being that due to the high number of measurements, only large areas of non-working channels may affect the pattern recognition and the muon trajectory reconstruction. The data set contains samples, where some isolated channels report extreme values. Hence, the one-dimensional median filter is applied:

$$\hat{x}_{i,j} = \text{med}(x_{i,j}, x_{i,j+1}, x_{i,j+2}) \, . \tag{4.3}$$

- *Normalization*: the absolute occupancy value of the chambers in the input data set depends on the integration time and the LHC beam configuration, and intensity, i.e. on the number of LSs spanned when creating the image and corresponding luminosity. The normalization strategy depends on the need for comparing data across chambers or runs: the precise procedure used in the two approaches is described in Sections 4.2 and 4.3, respectively.

4.1.1 Legacy Monitoring Strategy for DT Occupancy

Current monitoring strategy for DT occupancy plots is an excellent illustration of an approach widely used by CMS sub-detector communities.

The AD method used in the online monitoring production system targets a specific failure scenario: a region of cells not providing any electronic signal, large enough to affect the track reconstruction in the chamber. That is by far the most frequent issue, usually related to temporary problems in the readout electronics. Examples of this kind of failures are shown in Figure 4.1 B and C. These kinds of occupancy plots are created accumulating data in time. Once in a while, the plot filling process is reset, to increase sensitivity to problems occurring during the run. The layer expected behaviour is to report occupancy of hits with the small variance between adjacent channels. The legacy strategy evaluates samples per chamber and assembles them in so-called *summary plots*. In this manner, the human shifter has a broad overview of the sub-detector status in *one* plot instead of manually browsing over 250 histograms. Although the algorithm quantifies the fault severity based on the fraction of affected channels, it does not identify specific faulty layers or channels. The legacy strategy regards Figure 4.1 instance A as non-problematic, correctly classifies the chamber in Figure 4.1 B as anomalous, but it is not sensitive enough to flag the chamber in Figure 4.1 C.

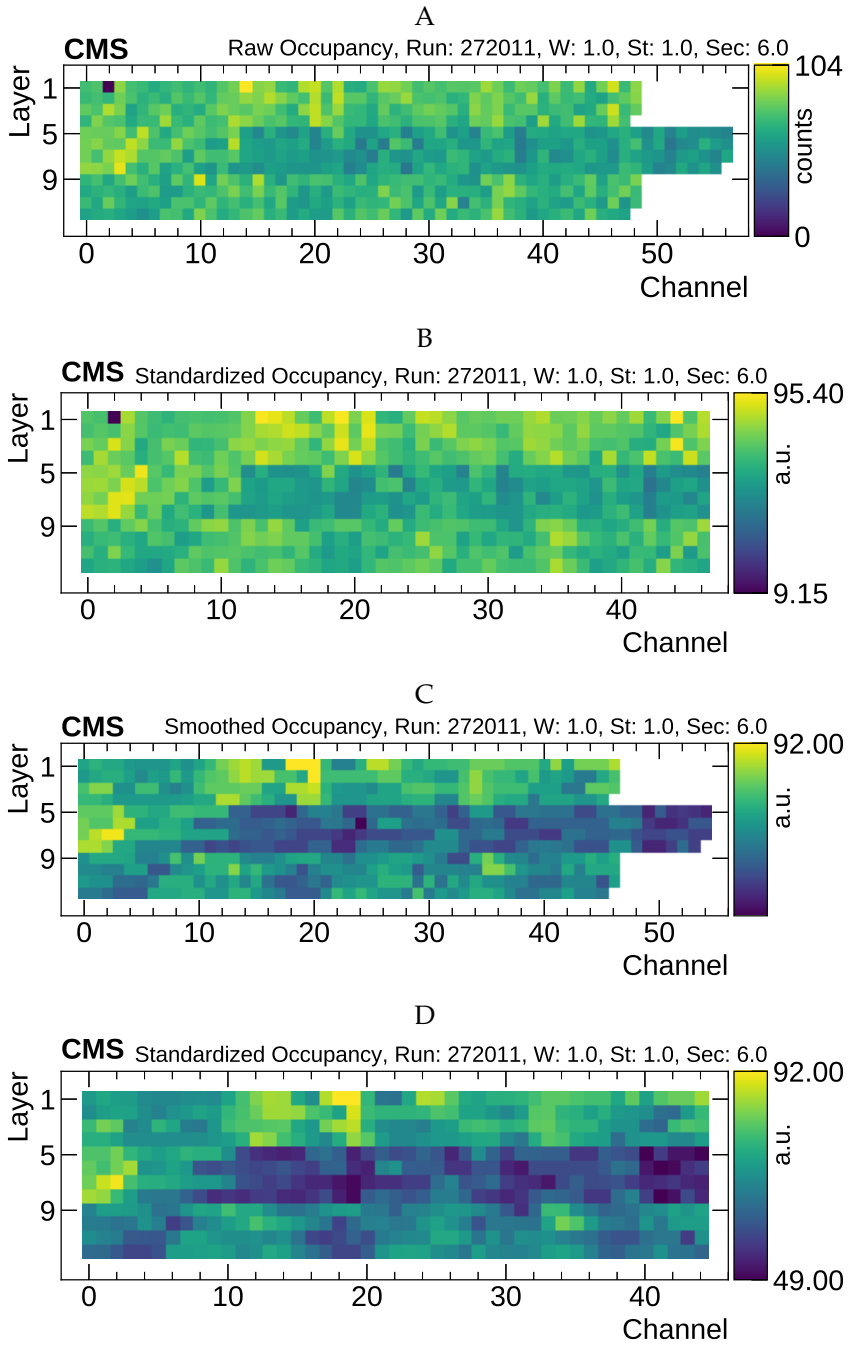


FIGURE 4.2: Example of two kinds of input sample preprocessing. (A) acquired (*raw*) values, (B) standardizing each layer directly from raw values using linear interpolation. (C) smoothing the raw values data with median filter (D) standardizing each layer from smoothed data. In (C), the isolated low-occupancy spot in layer 1, corresponding to a dead channel, is discarded.

Tests in other sub-detectors are approached analogously. The current level of automatization extends to the infrastructure that creates the plots and, the superposition to the existing reference. In several cases, some statistical test (e.g. Kolmogorov-Smirnov, χ^2) is performed, and the outcome is displayed. However, the interpretation and ultimate decision are taken by human shifter.

4.2 Identifying Known Failure Modes in Supervised Setup for the Local Approach

The first experiment concentrates on training a classifier to identify problems local to the DT layer: for this purpose data collected in each layer are treated independently from the others. This approach enforces the expert knowledge of what is currently considered correct or anomalous and probes the detector with higher granularity than the legacy strategy, i.e. per-layer instead of per chamber score. The goal is to identify regions of channels not registering any hits (called *dead channels* in detector jargon), or having lower detection efficiency (hence lower hit counts compared to the neighbouring ones in the same layer) or having an anomalously high hit count dominated by electronic noise (called *noisy channels*). These are, by far the most common failure modes.

The local approach can be considered as an initial benchmark comparing fully supervised, semi-supervised and unsupervised methods, and specific algorithms in each category, before embarking in full-fledged AD. Moreover, if successful, it can be further exploited as a preprocessing step for filtering these trivial faults before attempting to detect more elusive and novel ones.

Given the locality restriction of this approach, contextual information is not accessible. As a consequence, a model based on this strategy will not be able to spot, for example, a faulty layer in which occupancy is decreased uniformly compared to neighbouring layers in the same chamber. This limitation is addressed in Section 4.3.

4.2.1 Methods and Experimental Setup

After having applied the standardization procedure (see Section 4.1), a layer is represented as a single row of an occupancy matrix:

$$X_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, \ldots, \tilde{x}_{i,47}) \,. \tag{4.4}$$

The available data set consists of 21000 chambers corresponding to 228480 individual layers.

Hit counts in a layer are normalized to a $[0, 1]$ range, dividing them by the maximum of the occupancy value in the layer:

$$\dot{x}_{i,l} = \frac{\tilde{x}_{i,l}}{\max(X_i)} \,. \tag{4.5}$$

The need for normalization comes from the intrinsic variation of the occupancy, which depends on the spatial position of the chamber (as described in more detail in Section 4.4) and on the integration time of the analyzed image.

The ground truth is established by field experts on a random subset of the data set, by visually inspecting the input sample before any preprocessing: 5668 layers were labelled as good and 612 as bad. The 9.75% fault rate is a faithful representation of the real problem at hand. With this ratio, both anomaly and outlier detection approach can be considered. Out of this set 1134 good and 123 bad examples are reserved for composing the test set corresponding to 20% of the labelled layers. The remaining examples are used for training and validation for the semi-supervised and supervised methods.

Classical fully unsupervised approaches based on the neighbourhood (e.g. k-NN), topological density estimation (LOF and its variants) or clustering are not relevant here. These algorithms perform poorly in high dimensions, especially that a simple geometric (e.g. Euclidean) distance in the feature space does not define a similarity metric Zimek et al. (2012). For instance, the distance between examples A and B in Figure 4.1 is dominated by the contribution of well-behaving channels. The similarity function, or equivalently the adequate representation, must be learned from the data.

This representation learning view Bengio et al. (2013) points towards DL, as it should remain sensitive to the local geometric relationship in the data related to the underlying apparatus. CNN (see LeCun et al. (1995); Krizhevsky et al. (2012)) integrates the basic knowledge of merely the topological structure of the input dimensions and learn the optimal filters that minimize the objective error.

In this experiment, the performances of the following are compared:

- **unsupervised learning** with (a) a simple statistical indicator, the variance within the layer, and (b) an image processing technique, namely the maximum value of the vector obtained by applying a variant of an edge detection Sobel filter, see Sobel (1990):

$$S_i = \max(\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * X_i) \; ; \tag{4.6}$$

- **semi-supervised learning**, with (c) IF, and (d) μ-SVM; *and*
- **supervised learning**, with (e) a fully connected Shallow Neural Network (SNN), and (f) a CNN.

The IF and μ-SVM models are cross-validated using five stratified data set folds to search for their corresponding optimal hyper-parameters. Subsequently, the IF is retrained using those hyper-parameters (100 base estimators in the ensemble) on the full unlabelled data set, while μ-SVM (RBF kernel, ν of 0.4, γ of 0.1) is retrained using only negative class examples.

The architecture of the CNN model with one-dimensional convolution layers used for this problem is shown in Figure 4.3. Rectified linear units, see Nair and Hinton (2010), are chosen as activation functions for inner-layer nodes, while the softmax function is used for the output nodes. The model is trained using the Adam optimizer, see Kingma and Ba (2014), and early stopping mechanism that monitors the validation set (set to 20% of the data set) with patience set to 32 epochs. The model is implemented in Keras, see Chollet et al. (2015), using TensorFlow, see Abadi et al. (2016), as a backend.

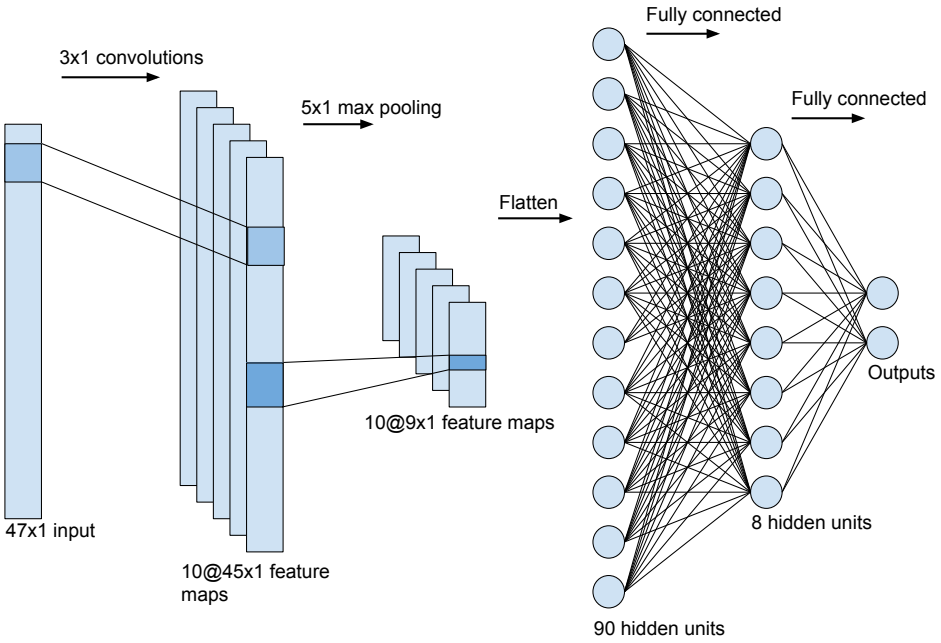


FIGURE 4.3: Architecture of the CNN model used to target the local strategy.

CNN was chosen because the problem at hand is naturally linked to image processing. CNNs can be interpreted as an automatic feature extractors from the image. In contrast to other traditional algorithms that may overlook spatial information between pixels, CNNs effectively use adjacent pixel information to extract relevant variability in data using self-thought filters and use a classification layer at the end.

The SNN model consists of one hidden fully-connected layer with 16 units (chosen to approximately match number of parameters in the CNN). As for CNN, it uses rectified linear unit as activation function of the hidden nodes and the softmax function is used for the output nodes. This model is primarily introduced to obtain a term of comparison for the CNN.

Unlike what was done for the other models, the smoothing preprocessing step described in Section 4.1 is not applied for CNN nor SNN models, in order to allow the networks to learn their filters. Additionally, the negative S_0 and positive S_1 samples are weighted to account for class imbalance. The weight λ_ψ for a sample in class $\psi \in \{0, 1\}$ is defined by

$$\lambda_\psi = \frac{|S|}{2 \cdot |S_\psi|}\,, \tag{4.7}$$

where $S = S_0 \cup S_1$.

4.2.2 Experimental Results and Discussion

The performance of the various models on a held-out test data set can be seen in Figure 4.4, which shows the different ROC curves. Compared to statistical, image processing or other ML-based solutions supervised DL outperforms the other models. Thanks to the limited

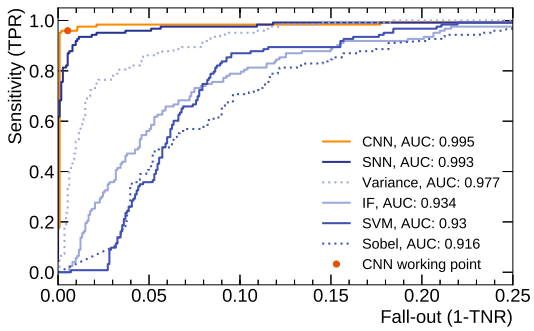


FIGURE 4.4: ROC curves for different models used in the local approach. The AUC is quoted to compare the performance.

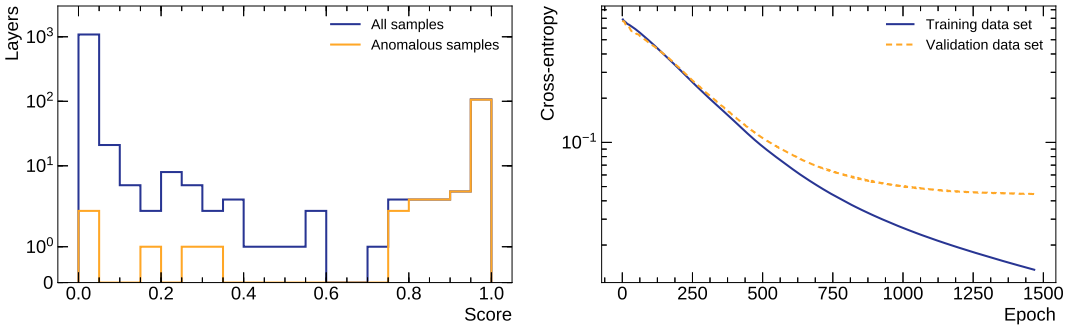


FIGURE 4.5: The distribution of scores (left), and the loss function as a function of the number of training epochs (right) for the CNN model used in the local approach.

number of parameters of the model, the training converges to a satisfactory result (Figure 4.5), even though the number of training samples was small.

Although the AUC of the fully-connected SNN is comparable to the one of CNN, the latter is a better solution when requiring maximum specificity (TNR, aims at avoiding FPs) and sensitivity (TPR, aims at avoiding FN).

The relatively good performance of the unsupervised variance method, compared to the poor results of the filter, and the near-optimal performance of the SNN, show that the filters to learn are not simple contrasts. However, the superior performance of CNN demonstrates that the initial edge detection layer is useful. Unfortunately, the filter visualization, see Figure 4.6 does not provide vital information for the human experts as the nature of the input data is much different than in the real-world data sets.

The limited performance of IF is likely to come from the violation of its fundamental assumption. The faults are not rare (remember that the fault rate is in the order of 10%) and

FIGURE 4.6: Visualization of the learned filters of the CNN model used in the local approach.

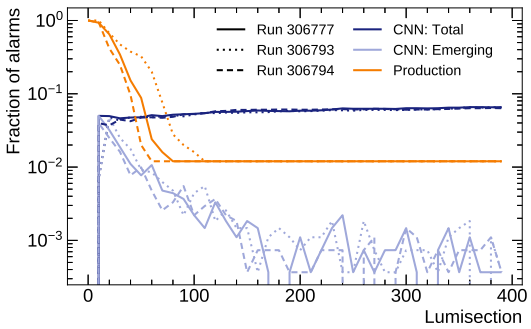


FIGURE 4.7: Stability of the proposed model and the legacy strategy as a function of time (*number of LSs*) for three different runs: 306777, 306793, 306794. The stability test is performed every 10 LSs. The *CNN: Total* and *Production* lines follow the total fraction of alarms generated by the methods. Instead, *CNN: Emerging* reports the fraction of new alarms being generated by the CNN model w.r.t previous test point.

homogeneous. The inferior performance of the typical semi-supervised method (μ-SVM) illustrates the well-known smoothness versus locality argument for DL, see Bengio and LeCun (2007); Bengio et al. (2013): the difficulty in modelling the highly varying decision surfaces produced by complex dependencies involving many factors.

As shown in the score distribution of Figure 4.5, the proposed architecture of the CNN model separates anomalous layers significantly. That allows for excellent flexibility in choosing the working point for deployment in production in the CMS DQM. Depending on the cost of type 1 and type 2 errors for the detector operators, the threshold can be set anywhere in $[0.1, 0.9]$ score range. When using the CNN for the selection of suitable samples for training the regional algorithms, the working point is chosen not to favour specificity nor sensitivity, i.e. the FPR and TPR have the same associated cost, with a threshold equal to score 0.5. For more information about cost matrix properties, see Elkan (2001).

The legacy strategy targets a specific failure scenario of dead regions without considering spatial proximity information. The algorithm produces a chamber-wise goodness assessment without being capable of identifying a specific problematic layer in the chamber. For this reason, a direct comparison with the CNN model is impossible. Instead, per-layer ground truth is used to label as bad any chamber with at least one problematic layer. The legacy algorithm is asked to indicate there is at least one faulty layer in a chamber. With this per chamber label, the specificity of the legacy strategy is estimated to 91%, with a sensitivity of only 26%.

Aside from higher sensitivity and specificity, CNNs offer an improvement in promptness of flagging emerging malfunctions. The performance with low statistics, i.e. at the beginning of a run, is a crucial area of improvement as all monitoring algorithms are built under the assumption of availability of high statistics. Thus, the early results at each run give high uncertainty and are ambiguous for the DQM shifter (in the context of the time). As seen in Figure 4.7, the CNN model gradually adds alarms until reaching stability (with most of the alarms being generated in the early stage of the run). The legacy strategy has the opposite behaviour, generating a substantial fraction of false alarms in the early stages of the run, gradually removing them. As a consequence, it has been reported that less experienced DQM shifters have incorrectly interpreted those results as a significant failure in the detector.

4.2.3 Interpretation of Classification Results

Along with public discord on the interpretability of neural networks, especially CNNs, comprehensive research has already been done to understand network predictions and classifications, see Montavon et al. (2018) for an overview.

As discussed in length in Simonyan et al. (2013), getting insights into the DL decision-making process could be done by either investigating class model visualization or class saliency in an image. Both of them require computing the gradient of the output for the input image. The first method generates an image that best represents a given category (class). The latter finds the pixels of an image which contribute most towards a particular decision. It is calculated by taking the derivative of the class score for the input image space is taken and evaluated for a given image. Given an image I_0 a class score $S_c(I)$ is approximated with a linear function in the neighbourhood of I_0 by computing the first-order Taylor expansion:

$$S_c(I) \approx w^T I + b\,, \tag{4.8}$$

where w is the derivative of S_c for the image I at the point I_0:

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}\,. \tag{4.9}$$

That results in a scalar quantity for each pixel in the image.

Example of saliency maps visualization generated for DT occupancy matrices is shown in Figure 4.8. The channels with high values match the regions of low occupancy. These plots were proven fundamental to point the detector experts to the root of the CNN decision allowing them to carry on further investigations on the detector aspects. Also, in case of incorrect classification, the saliency maps could be used to understand the decision of the algorithms in detail and take corrective measurements.

Saliency map can also be used for isolating objects of interest. More recent work on object detection may be more suitable for this task, e.g. Redmon et al. (2016). Alternatively, instead of using back-propagation of the output class score for the input Zhou et al. (2016) proposes to modify the network architecture so that the forward propagation can perform both the classification and localization. In the future, the experiment experts may find localization tasks necessary for further automatization of CMS DQM. In such a case, one can consider the state of the art techniques and, e.g. Selvaraju et al. (2016); Bojarski et al. (2016); Sundararajan et al. (2017). At this stage, proposed saliency maps are a perfect balance between simplicity of implementation and interpretability of classification results.

4.2.4 Model Improvement with Active Learning

Annotating hundreds (for simpler models) or thousands of histograms for ML-based CMS DQM can be tedious or even redundant exercise.

Active learning (also known as query learning or optimal experimental design) is an area of research particularly suitable in cases when the data collection is a cheap process while the data labelling is not (e.g. time-consuming). It follows a simple yet powerful idea of querying an oracle (i.e. human expert) for a label for most informative samples. There are several

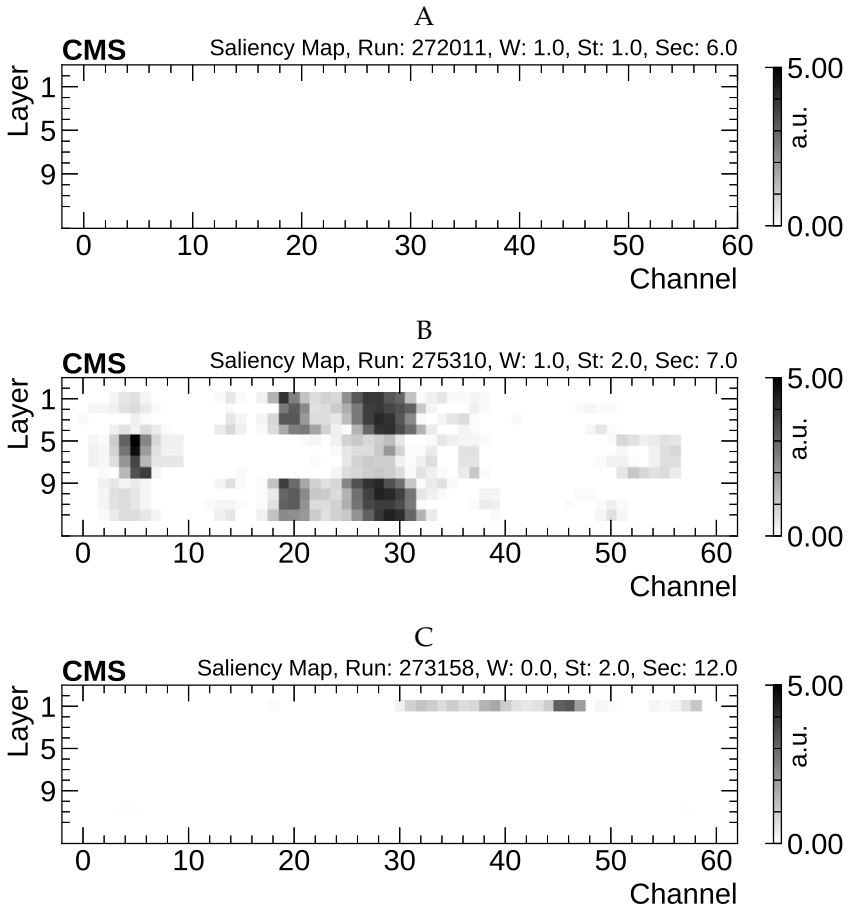


FIGURE 4.8: Example of visualization of saliency maps for three DT chambers corresponding to input occupancies from Figure 4.1. The scale is proportional to the channel influence over classifier decision to flag problems.

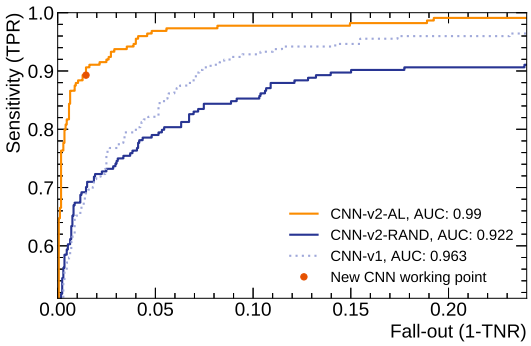


FIGURE 4.9: ROC curves for different versions of CNN. CNN-v1 is the model trained on the initially labelled data set. Its AUC drops compared to Figure 4.4 as the test set is extended with new samples to ease comparison between different versions. CNN-v2-RAND and CNN-v2-AL are models trained on a set $\sim$ 20% bigger than the original one with additional labels coming from different sources. CNN-v2-RAND uses labels coming from a random selection of chambers while CNN-v2-AL uses uncertainty sampling. The experiment shows that the incorrect choice of samples to label can lead to a drop in model performance.

scenarios in which active learners may pose queries and several query strategies that have been used to decide which instances are most informative. For a general survey, see Settles (2009). Despite the consensus on active learning and its widespread use, the literature is reporting both positive Tomanek and Olsson (2009) and negative Schein and Ungar (2007) results.

The most frequently seen scenario of active learning is a pool-based setting. The queries are selectively drawn from the pool of full data set based on the knowledge of its contents. Alternatively, in a selective sampling setting, see Atlas et al. (1990), the algorithm decides on the informativeness of each sample independently, on a one-by-one basis given a criterion known upfront. That method is also referred to as *stream-based sampling*. Lastly, in another setting, the model can be trained with generated, informative samples, known as query synthesis.

The most commonly used query strategy is uncertainty sampling. In this framework, an algorithm queries an oracle for the label for the instances about which is least sure. For example, when using a probabilistic model for binary classification, uncertainty sampling simply queries the instance whose posterior probability of being positive or negative is closest to 0.5, see Lewis and Gale (1994); Lewis and Catlett (1994). For binary classification, other margin-based and entropy-based selections are again equivalent to querying the instance with a class posterior closest to 0.5. The alternatives such as Query-By-Committee discussed in Seung et al. (1992), Expected Model Change, see Settles et al. (2008), Expected Error Reduction, see Roy and McCallum (2001), require a more complicated setup.

For online DQM, a simple pool-based setting is a correct choice as the data set is available upfront, and there is no need nor feasibility for labelling instances on-the-fly. Synthesized samples may look confusing to the oracle: a lack of physics-based rules during synthesis will produce unanticipated, unrealistic histograms. The experiment using uncertainty sampling for the AD for DTs local approach was undertaken. The results are shown in Figure 4.9 and suggest the implementation of a pool-based uncertainty sampling active learning for the local approach as it improves classification results.

Above all, some aspects have to be addressed, which for now are inconclusive. First and foremost is the question of batch size before retraining (amount of new labels to generate in each cycle). The detector experts are advised to vary the amount of labelling to leverage between the cost of the process and the observed model inaccuracy reduction. Finally, the stopping criterion needs to be defined by the sub-detector communities.

4.3 Relative Comparison of Detector Components for the Regional Approach

In normal conditions, healthy chambers show similar occupancy levels in adjacent layers with the four inner layers having a different behaviour due to their different orientation (see Section 2.3.2). The regional approach described in this section exploits the pattern of relative occupancy of the layers within a chamber. In essence, this approach extends and complements the local one, allowing to identify less frequent intra-chamber problems which require the comparison of the information about all layers within one chamber in order to be spotted. For example, it aims at detecting failure modes where the occupancy of hits decreases uniformly in a specific layer or set of layers.

Typical examples of these kinds of failures are problems related to the high-voltage bias of the drift cells. The voltage distribution system is organized by layers and a lower value w.r.t to the nominal operation point would result in lower detector efficiency and, as a consequence, lower absolute occupancy in the affected region. Figure 4.10 shows an example of such an occurrence, where layer 9 is misbehaving. The legacy strategy and the local models of Section 4.2 are not conceived to detect this type of anomalies. All layers in a chamber are considered simultaneously to solve this challenge; each chamber is considered independently from the others.

4.3.1 Methods and Experimental Setup

In the early stages of this work, it was observed that a model capable of detecting regional anomalies could not be successfully trained if the local faults are not filtered beforehand. For this purpose, the circa 500 labelled images available for this study does not provide a sufficiently large training set. Thus a much larger data set is used (all the unlabelled samples). The pre-filtering problem is solved using the score of the convolutional model presented in Section 4.2 as an approximation of the ground truth. For this, a working point with 96% TPR is picked, corresponding to a 1% FPR (to guarantee an extensive data set size and low level of contamination). The amount of problematic chambers is reduced to a tolerable level, comparable to a human expert.

All chambers with any layer identified as faulty are discarded. Chambers located in MB4 are discarded as well, because of the lack of a middle group of four layers, see Section 4.1. The above changes effectively narrowed the training data set to 8441 matrices. The smoothing and standardization procedures are applied to all the layers $\tilde{C}^k$ within each chamber obtaining matrices of shape 12×46. The occupancy of hits within one chamber are normalized

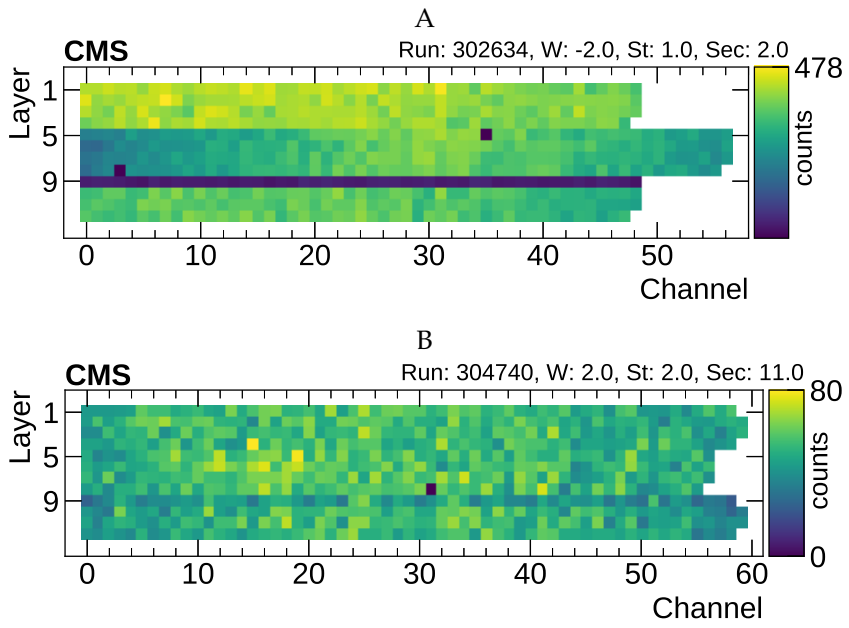


FIGURE 4.10: Example of the impact on hit counting of different voltages applied to layer 9. (A) shows the occupancy map when operating the layer at 3200 V and (B) shows the effect of operating at 3450 V. Both examples should be regarded as anomalies. Since the values in both cases are not equal to zero, the legacy strategy considers those cases as non-problematic.

using a min-max scaler:

$$\dot{C} = \frac{\tilde{C}^k - \min(\tilde{C}^k)}{\max(\tilde{C}^k) - \min(\tilde{C}^k)} \,. \tag{4.10}$$

This normalized values to the $[0, 1]$ range while retaining the information about the relative occupancy between the layers.

In order to evaluate the model, the only labelled set for the class of anomalies to tackle: a subset of the data during which layer 9 of some chambers were operating at a voltage lower than the nominal one (see Figure 4.10). In particular, the voltage was set to 3450 V[1] and 3200 V[2] while the standard operation point is 3550 or 3600 V depending on the chamber. These settings result in an absolute difference in hit counting, more pronounced for the lower voltage settings because of the physics of gas ionization by radiation. The chambers where all layers operate at nominal conditions are considered as *good* in the test.

In this experiment, the following unsupervised methods for semi-supervised AD are considered:

- simple undercomplete autoencoder with the 20 unit representation layer;
- convolutional autoencoder;
- denoising autoencoder where with additional noise in training samples; *and*
- autoencoder with kernel L1 (10^{-5}) sparsity regularization in the hidden layers.

Similarly to the local approach, the autoencoders are trained using Adam optimizer. Early stopping mechanism with the patience set to 32 epochs is adopted to monitor validation

[1] for CMS members: runs 304737-304740
[2] for CMS members: run 302634

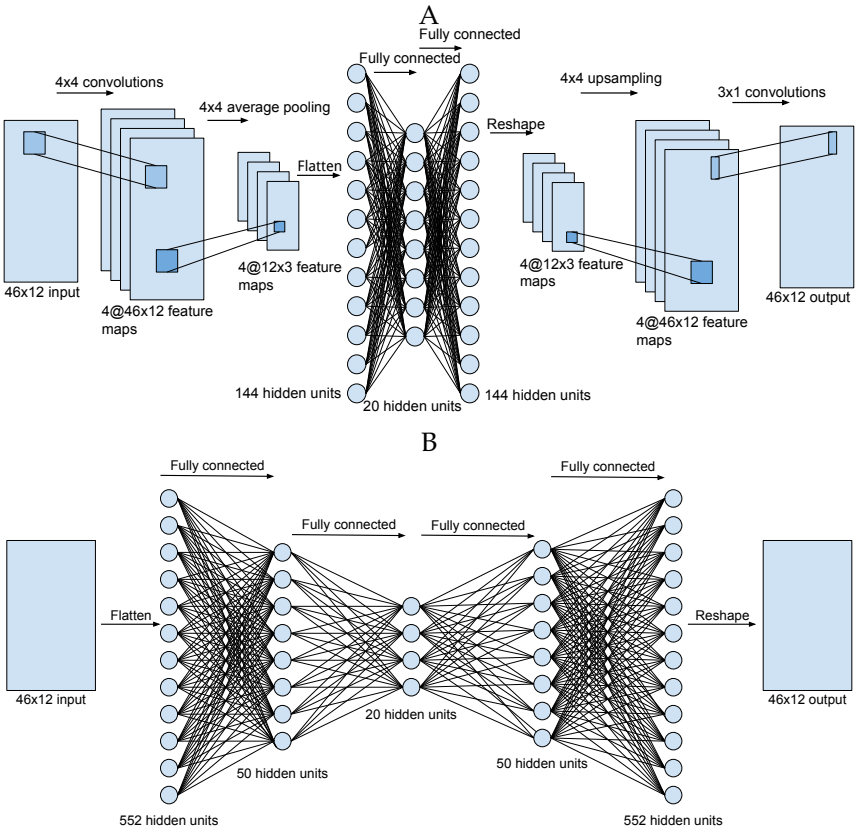


FIGURE 4.11: Convolutional (A) and simple, denoising, sparse (B) autoencoder models architecture used to target *regional* strategy.

set (20% of the total data set). All models are implemented using the Keras library with TensorFlow as a backend. The architecture of the model is shown in Figure 4.11. A and B for, respectively, the convolutional autoencoder and the other three models (for which a base architecture is adopted). The bottleneck architecture is kept for both denoising and sparse autoencoders to limit the number of training parameters. The parametric rectified linear unit is used as the activation function on the hidden layers, while the output layer uses the sigmoid function. All models are instructed to minimize the Mean Squared Error (MSE) ϵ between original, $\dot{x}$, and reconstructed, $\ddot{x}$, samples:

$$\epsilon^k = \frac{1}{ij} \sum_{i,j} (\dot{x}^k_{i,j} - \ddot{x}^k_{i,j})^2 \,. \tag{4.11}$$

4.3.2 Experimental Results and Discussion

The following quantity as anomaly indicator is taken:

$$\epsilon^k_i = \frac{1}{j} \sum_{j} (\dot{x}^k_{i,j} - \ddot{x}^k_{i,j})^2 \tag{4.12}$$

to assess the performance of a given ensemble of channels, i.e. the MSE between the original sample given as input to the encoder ($\dot{x}^k_{i,j}$) and the output of the decoder ($\ddot{x}^k_{i,j}$). The granularity of the autoencoder information is exploited to identify the problematic region of the

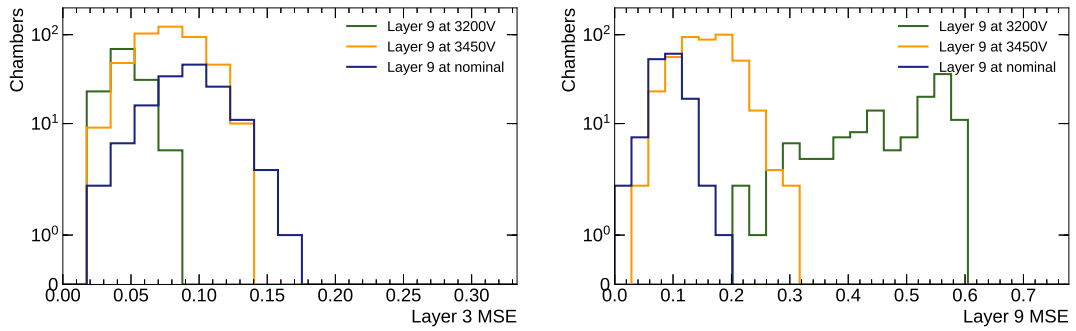


FIGURE 4.12: MSE between reconstructed and input samples for layer 3 (left) and layer 9 (right) for three categories of data for convolutional autoencoder. Despite a problem in layer 9, all ϵ for layer 3 are comparable for all chambers. The nominal voltage falls between 3550 and 3600 V .

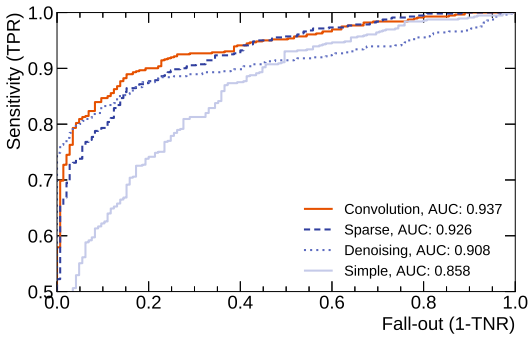


FIGURE 4.13: ROC and AUC of the different autoencoder models used in the regional approach. The anomaly score for the test set is the MSE in layer 9. See Figure 4.12 for comparison of anomaly scores.

chamber by computing the MSE values for a different set of channels. For example, one can compute the MSE for all the channels corresponding to a given read-out electronic board or computed per layer when tackling potential failures of the voltage distribution system. On the one hand, this allows experts to feed their technical knowledge into the evaluation process but also suppress FP as most of the distance between autoencoder input and output come from well-behaving channels.

This figure of merit is used for the sample with different voltage settings. Figure 4.13 shows the excellent performance of all models, especially convolutional autoencoder. The distributions of the MSE for a well-behaving and a problematic layer are shown in Figure 4.12. The MSE distribution for layer 9 shows clear separation for chambers operated at nominal and lower voltages. For each ϵ_i value, a quantitative assessment of the severity of a potential anomaly can be derived quoting the corresponding p-value of the excellent example distribution. The separation is not evident for the working point at 3450 V, which is closer to the standard setup, reflected in the AUC values reported in Figure 4.13.

The legacy strategy is not sensitive to the type of faults described in this section since the hits in layer 9 are non-zero values. For deployment in the DQM infrastructure of the CMS experiment, the local and regional models can be applied in a pipeline to cover all different kind of anomalous behaviours.

4.4 Identification of Emerging or Novel Problems for the Global Approach

The global approach aims at detecting anomalies looking at the global ensemble of muon chambers, exploiting the dependency of the occupancy of each of them on their position in the detector.The chambers are categorized according to their position in the spectrometer and the occupancy pattern, exploiting the field knowledge to predetermine the chambers classes. The model is fed with the occupancy data of all chambers for a given run and provide a tool to visually investigate for a novel or emerging problems through dimensionality reduction.

The position of the chamber in the CMS detector (uniquely determined by the wheel, station, and sector numbers) impacts expected occupancy distribution of the channel hits. The expected occupancy pattern is mainly driven by the proximity to the beam-collision point, at the centre of the detector. As a consequence, chambers in different stations (see Section 2.3.2) will manifest a different behaviour. The rotational symmetry of the detector geometry and the collision events around the beam axis is taken into account grouping chambers within the same station, independently on the sector they belong to (see Section 2.3.2). Similarly chambers belonging to the same station but in opposite wheels are considered alike. Additionally, the behaviour of the chambers is expected to be the same across different runs (modulo the decrease of occupancy due to the decrease of beam intensity across the fill). That leaves us with a categorization based on the chamber numbering schema, where the station number and the absolute value of the wheel number are the only relevant parameters.

The problem is contextual, in the sense that critical explanatory attributes are not part of the underlying data features. Conditional AD, see Song et al. (2007), has been proposed to deal with such a situation when the relevance of external attributes is unknown. For instance, if a set of environmental or technical attributes are monitored, that can impact the behaviour of the detector components. The spatial positions of the chambers are the only external attribute, and the common understanding of the underlying physics processes assures their impact. Thus, the problem is back to a point anomaly problem.

4.4.1 Methods and Experimental Setup

In this approach, undercomplete autoencoder is used, similar to that introduced in Section 4.3 (see Figure 4.11), except that the size of the representation layer is reduced to three units for visualization purposes. The same preprocessing, training and validation procedure is followed.

The goal is to exploit the categorization of the chambers based on their geographical location to interpret the compressed representations.

At the moment of writing this work, global faults are not systematically labelled by the DT detector experts. In the absence of a global label, only an unsupervised method and visual interpretation are considered for this experiment.

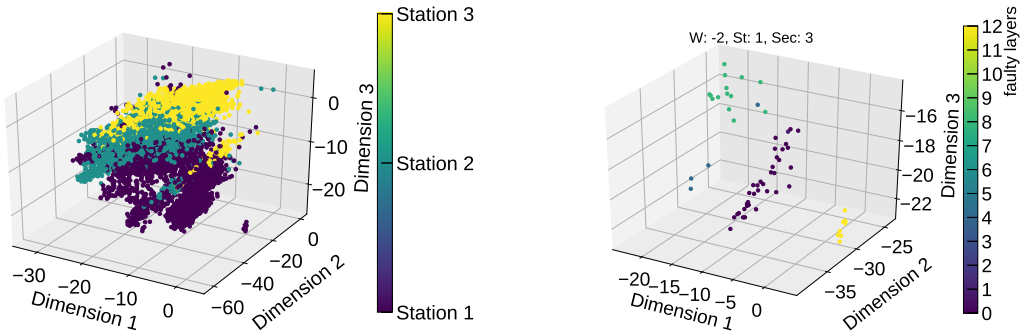


FIGURE 4.14: Compressed representations of chamber level data for all chambers (left) and limited to only one chamber across different runs on a scale of the number of anomalous layers (right). The samples cluster according to their positions in the detector, i.e. station number (left), or according to similar behaviour (right).

4.4.2 Experimental Results and Discussion

Figure 4.14 shows an example of a low-dimensionality representation of the chamber data clustering depending on the chamber position in the detector. The global approach is then capable of spotting an unusual behaviour of DT chambers, taking into account the geographical constraints. Ultimately, this could pave the way to more flexible assessment by scoring per detector region.

When investigating the representations for a specific chamber across different runs (see Figure 4.14), the representations tend to cluster depending on the number of problematic layers. Thanks to this fact, the cumulative distribution of the compressed representation could be used to highlight the occurrence of new anomalies or to associate an anomalous behaviour to an already known problem. This application could assist experts in diagnosing transient and reoccurring issues.

4.5 Conclusions and Practical Considerations

This chapter describes how detector components malfunctions can be identified with high accuracy by a set of automatic procedures, based on ML AD. The specific case of the DT muon chambers of the CMS experiment was considered. A CNN-based classifier is proposed to spot local misbehaviours of the kind currently targeted by the existing CMS monitoring tools. The classifier is capable of detecting the known anomalous behaviours with unprecedented efficiency. It is possible to extract more information from the map of electronic hits than what is achieved by the legacy strategy. In particular, a strategy to spot regional problems across layers in a detector chamber is proposed, or globally, i.e. across chambers in the full muon detector using convolutional autoencoders.

The algorithm for the detection of anomalies within the layer (local approach) has been integrated into the CMS online DQM infrastructure at the beginning of the 2018 LHC Run and kept running in parallel with the legacy strategy. That allowed to commission it using the newly acquired collision data. After initial tuning of the working points to meet the requirements of the DT detector experts, the algorithm has been performing reliably, and it is

considered for deployment in the next LHC Run (2021). The deployment and commissioning of the regional and global approaches are also foreseen in the future.

A generalization of these strategies paves the way to the full automation of the data quality assessment process for present and future HEP experiments. Since CNN is the primary ingredient in this study, and since many monitored quantities in typical HEP experiments are based on 2D maps (e.g., detector occupancy, detector synchronization), the approach proposed in this paper could be extended beyond the presented use case. A study set for Gas Electron Multiplier (GEM) detectors, see Kang and Lee (2018), confirmed this claim.

A web-based framework for data labelling and algorithm re-training was developed to facilitate the process of adopting a similar strategy for different sub-detectors. This web-based solution serves as a feedback to expert field knowledge in the automated test suite. It allows for automated acquiring and labelling of a large amount of data. It assists experts in debugging networks decisions by visualizing saliency maps using gradient backpropagation and facilitates continuous improvement of the models by implementing pool-based uncertainty sampling for active labelling.

Based on the results presented in this chapter, the novel approaches provide better and more scalable solutions when compared to the classical approaches. As the neural networks are universal approximates, they deal with nuances and non-linearity of the input space efficiently. Classical AD usually deals with smaller input dimensionality. For data sets with high dimensions, DL comes as an easily trained alternative. The standard caveat of the necessity of obtaining a large amount of training data for supervised approaches could be overcome with the low number of training parameters of the model. Finally, apart from CNNs ability to classify real-world images correctly, they can deal with images composed by the monitoring infrastructure (local approach).

The autoencoders can be used for semi-supervised AD tasks. Algorithms based on autoencoders, offer a more robust AD strategy, not being defined as supervised classifiers of specific failure modes. This approach allows localizing the origin of a given anomaly, exploiting the granularity offered by the use of MSE of the decoded image as a quantification of the anomaly.

CHAPTER 5

Data Certification Novelty Detection with Deep Autoencoders

The data acquired by the CMS experiment are scrutinized by a certification procedure, which ensures they are usable for all physics analysis, see Schneider (2018). This procedure is the last step of the complex DQM apparatus of the experiment, described in Section 2.4. The current procedure is conducted by experts of the various sub-detectors and is based on monitoring some fundamental physics quantities computed at various stages of the data-processing chain (see Section 2.4.3).

An efficient ML algorithm requires reuse of the current knowledge about the world to take on a new, possibly anomalous observation. To this purpose, the algorithm needs to understand the similarities between old and new samples. Human experts make decisions on the data quality based on statistical distributions displayed in the form of histograms produced by the DQM infrastructure. These distributions are used in two ways. On one side, the field experts apply their prior knowledge of the physics processes and the detector functioning to judge what is expected and what is not. On the other side, they compare consecutive chunks of data looking for striking and unexpected changes. This chapter tackles these two aspects by exploring the use of autoencoders to target novelty detection. The main challenge is to train a model that learns the features of a specific data set and discovers anomalies in the data. To ensure adequate coverage of all possible physics signatures of interest to the physicists, the model has to probe a large number of features. At the same time, the procedure needs to ensure that variations in the quality of the data are identified with the best possible time granularity: this requires being able to probe data corresponding to a short period of acquisition and hence with limited statistics.

The CMS data, as well as the DQM task, are organized in *acquisition runs* corresponding to several hours of data taking (see Section 2.3.4). In case an anomaly is detected by the certification procedure, the corresponding data need to be discarded. The exact time intervals affected by the anomalous behaviour need to be correctly identified to minimize the data loss. That requires the certification procedure to be sensitive to transient effects. When the affected interval of the acquisition run is short, the statistics available in the DQM histograms are often too shallow for human assessment.

As a consequence, transient problems are challenging to identify and very often, a conservative approach has to be adopted discarding more data than actually necessary. In these cases, pinpointing the exact time interval affected by anomalous behaviour can require further investigation and the use of non-event data. A certification protocol based on a shorter interval, using luminosity sections (LSs, see Section 2.3.4), is more desirable. LS-based quality labels are already in place and are usually set using log messages and monitors related to the detector slow control more than on the data themselves, see Rapsevicius et al. (2011). The ever increasing physics data volume, as well as detector complexity, calls for ways to further automate this monitoring step. The CMS Collaboration is looking into new algorithms allowing it to assess the quality of the physics objects in the reconstructed data with high accuracy and low time granularity.

The data acquired by the experiment are subdivided into several data sets depending on their physics content. Each data set undergoes a reconstruction procedure yielding several different collections of physics objects. The certification procedure needs to assess the performance of all of them: this high data dimensionality naturally points toward the exploration of DL algorithms. The anomalies can be caused by detector malfunctions or sub-optimal software reconstruction and, by nature, are rare and not known *a priori*. Consequently, the use of supervised AD methods, such as binary classification neural networks, is problematic as a positive (anomalous) class may be misrepresented in the training set. Furthermore, the characteristics of *good* data are evolving with the LHC or CMS configuration.

A vital requirement of this approach is that the reason for flagging a sample is easily explainable, which can be further used to ascribe the origin of the problems in the data to a specific sub-detector or physics objects. This critical aspect would allow the collaboration for not only certifying the data but further improve the infrastructure.

This chapter introduces new tools for targeting the automation challenges illustrated above. This tool is based on a semi-supervised approach which uses deep autoencoders, see Goodfellow et al. (2016), trained on the data acquired during the 2016 LHC campaign. In summary, the main aspects of this work are:

- detecting different types of anomalies affecting the CMS detector with high sensitivity and specificity using in training only the knowledge about the good-quality data and no assumption on the type of anomaly;
- exploiting the statistical behaviour of 1-dimensional distributions of a large number of physics quantities;
- assessing the (*mis*) behaviour based on a shorter interval than what is realistically feasible with current protocol;
- exploiting semi-supervised models to achieve stable performance over time despite the natural evolution of the underlying physics quantities due to the LHC and CMS data-taking evolution;
- allowing for fine-grained interpretation of the classification results, which can be further used to ascribe the origin of the problems; *and*
- providing an additional tool in CMS DQM toolbox that minimizes the risk of human mistakes and speeds up the certification procedure.

5.1 Data Set and Preprocessing

After the data are recorded, a complicated data processing step (called physics *reconstruction*) transforms the output of electronic read-out signals into human-interpretable variables ascribed to the physics objects (e.g. photons, muons, jets, tracks). These data are persisted in the so-called Analysis Object Data (AOD), providing the physicists with a compact representation of the physical processes in a convenient format for the analysis processing. The present study is based on this AOD data format, considered as the best trade-off between the accuracy of the description and compactness and volume of the data. Other options would be to include more low-level quantities related to the detector performance and read-out granularity (e.g. as the ones used in Chapter 4 for the DT detector monitoring) or to further reduce the data volume going to more streamlined representation of the physics process as done in previous research work from Borisyak et al. (2017).

The data set used in the current work consists of all 163684 LSs data recorded from June to October 2016. Several types of reconstructed particle objects are included to maximize the coverage of the algorithms for different physics objects and possible anomalies. In total, 401 physics variables are used (e.g. transverse momentum, energy, multiplicity, direction for the different physics objects). Whenever the ground truth is needed, the quality labels (*good* or *bad*), determined by the manual certification procedure performed by the detector experts can be used.

Naturally, given the heterogeneity of the monitored quantities, the features follow different distributions spanning very different ranges. The data are preprocessed to speed up the training of the model: each feature is standardized by subtracting the mean and scaling it to unit variance.

As explained earlier, human experts make decisions regarding the data quality based on the shape of the statistical distributions of key quantities represented in the form of histograms. In case of an anomaly, the corresponding histograms should show a considerable deviation from the nominal shape. To mimic this logic, the distribution $D_i = \{x_0, \ldots, x_k\}$ of each one of the 401 monitored variables is represented by its summary statistics using five quantiles, mean and standard deviation (for visual interpretation see Figure 5.1). The final vector has 2807 features. Each data-point represents the data acquired during one LS to aim for high time granularity of the classification results.

5.1.1 Different Event Topologies

As mentioned before, the physics data is stored in different Primary Datasets (PDs). PDs are subsets of the event stream acquired by the CMS experiment grouped according to the presence of different types of particle candidates. The current DC process uses a subset of the PDs chosen to guarantee coverage of all the main physics objects, i.e. SingleMuon PD for *muons* or EGamma PD for *electrons*. Initial study uses a data set defined by the presence of hadronic showers, called *jets* in reconstructed collision products; the signature of jets involves all of the CMS sub-detectors, enabling this research to cover as many aspects of the detector as possible. However, the proposed strategy is generic enough to be extended to different PDs in the future, to guarantee complete coverage of the physics signatures. Ultimately, one could use 21 independently trained classifiers, each specializing in the classification of each

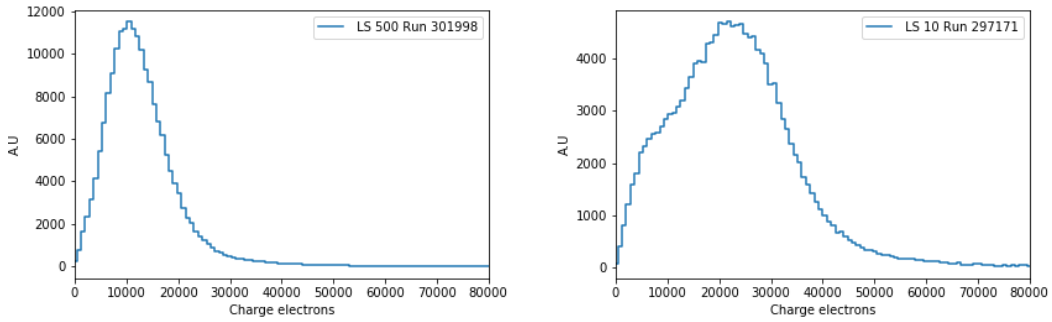


FIGURE 5.1: Two examples of histograms related to the Pixel detector (see Section 2.3.2) status for a normal (left) and anomalous (right) LSs. The reference shape is the Landau distribution. The bad LS manifests anomaly in low charge, which is caused by the Pixel detector not being properly synchronized with the bunch crossing. Such distributions, obtained for each LS, are preprocessed into a summary statistics vector of seven variables: five quantiles, mean and standard deviation.

PD, similarly to the architecture proposed in Azzolini et al. (2018). The global status of the detector could then be derived as a logical `AND` of the intermediate results coming from each of the models. While these aspects are crucial for the deployment aspects of the proposed method, the current work focuses on prototyping the strategy.

5.2 Semi-Supervised Novelty Detection with Deep Autoencoders

A critical requirement for a well-performing supervised algorithm is its ability to generalize to unseen observations. Such methods rely on big sample size during training and are vulnerable to incomplete or inadequate representations of data points in each class. In the continually changing environment, the implementation of these algorithms is thus more challenging as the incoming samples may considerably differ from the training data. As a consequence, in case of such novel data points, the algorithm is likely to classify them as one of the classes, perhaps good, while the expected output should be *other*, or *novel* class.

Novelty detection is the task of highlighting data that differs substantially from the past. The problem could be approached as an OCC task, as summarized in Chapter 3.1. In such a case, the algorithm is expected to define a boundary around the negative class $p(X) < \alpha$, and declare all test points outside the threshold α as novel ones. Classical OCC methods, such as μ-SVM do not scale well to large high-dimensional data sets.

An alternative and practical strategy could be to generate synthetic novel samples, and the problem will be then reduced to a fully supervised one. However, such a solution requires accurate forecasting of potential failure modes.

Anomalies are rare in the CMS data: they account for roughly 2% of the data set, which is a small set of examples of failures. Moreover, emerging, unprecedented failures are difficult to anticipate. Thus supervised methods may be deficient in the DC case. An alternative is a Semi-Supervised Novelty Detection (SSND) approach that models only negative (good)

class distribution, as a form of OCC. During data taking, the model aims at identifying unobserved patterns in newly recorded data. In this manner, the full potential to catch all the future and unseen detector failure modes is retained.

5.2.1 Methods and Experimental Design

The work in Blanchard et al. (2010) discussed SSND and proved that the presence of a mix of labelled negative and unlebelled samples, it is possible to obtain an optimal novelty detector. However, classical OCC usually deals with smaller input dimensionality. For data sets with high dimensions, DL comes as an easily trained alternative. The autoencoders are exploited for SSND to the purpose of the application at hand, under the assumption that, when trained on a negative class, they yield sub-optimal representations for novel samples even in the presence of label contamination. Similarly to intuition in Chapter 4, the discrepancy between input and the output of the neural network should indicate that a sample is likely generated by a different (novel) process.

The autoencoder architecture shown in Figure 5.2 is used with different regularization techniques. These includes:

- undercomplete autoencoder;
- contractive autoencoder, see Rifai et al. (2011);
- sparse autoencoder, see Ranzato et al. (2006); *and*
- variational autoencoder, see Rezende (2014).

Although the ML Community has proposed a variety of different regularization techniques, the above are the most basic and well-established techniques, as discussed in Bengio et al. (2013). The undercomplete autoencoder, which uses the only restriction in a bottleneck, serves as a baseline for other methods. In order to go beyond simple dimensionality reduction while preventing over-fitting the contractive, see Rifai et al. (2011), and sparse, see Ranzato et al. (2006), autoencoders are tested. A more ambitious goal is to extract an explanatory representation of the novelties with latent variables, in a probabilistic framework: the variational autoencoder, see Rezende (2014); Kingma and Welling (2013). For a general overview of these methods, see Chapter 3.4. This comparison will allow for assessing the hypothesis that autoencoders are indeed suitable for novelty detection and the standard regularization techniques are sufficient for helping with detection performance.

The sparse autoencoder has additional $L1$ kernel regularization (10^{-5}) on all of the hidden nodes. This constraint penalizes the output of the hidden unit kernels and forces them to be close to zero. The exact penalty term was established using a random search. For contractive autoencoder, additional regularization should improve model robustness against small variations in the training examples. Lastly, a VAE was tested (see Section 3.5). Because input values are not scaled to the predefined range, a parametric rectified linear unit He et al. (2015) is used as an activation function in the output layer. This way, the network learns the correct coefficient of leakage for a non-zero gradient when the unit is not active. Hidden units are also using this type of activation. The network is trained with Keras, see Chollet et al. (2015), and TensorFlow, see Abadi et al. (2016), using the Adam optimizer (with a learning rate of 0.0001, $\beta_1 = 0.7$, $\beta_2 = 0.9$), see Kingma and Ba (2014), and early stopping

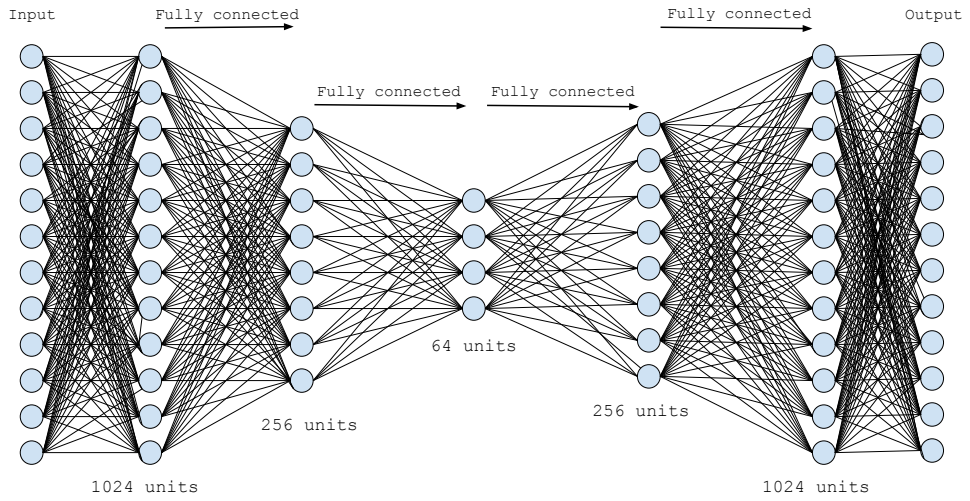


FIGURE 5.2: Proposed base autoencoder architecture for DC novelty detection. The hyper-parameters were chosen using a grid search.

mechanism monitoring validation data set with patience set to 32 epochs. The network is instructed to minimize MSE between input X and the output $\hat{X}$ vector:

$$\epsilon = \frac{1}{n}\sum_{i=0}^{n}(x_i - \hat{x_i})^2 \,. \tag{5.1}$$

The DC is performed on the newly acquired data while the data taking is ongoing: the detector experts do not have the complete data set at hand but only the portion acquired up to that point. The algorithm evaluates the LSs in the same order the apparatus have originally recorded them to mimic this *modus operandi*. The data set is split into training (60%), validation (20%) and testing (20%) sets after sorting all samples chronologically to simulate this production scenario. Since both the LHC and the response of the CMS subdetectors evolve gently with time, random splitting could lead to unintended data snooping, see White (2000), where the model is tested on LSs nearly identical to ones in the training sets. It was noticed that the anomaly contamination in the training set harms the performance of the algorithm. Thus all the bad samples are removed from training and validation sets. The test set is extended by those anomalous samples previously removed. Including more positive examples in the test is a better approach, as the original set has a limited amount of them. This approach helps qualify the performance of various methods given that bad LSs should always be qualified as anomalous.

The difference between reference and recorded distributions is dominated by noise. Experts pay attention only to significant deviations. The final decision function must mirror this behaviour, and it is computed using MSE of only the worst 100 autoencoder reconstructed features (TOP100):

$$\text{TOP100} = \frac{1}{100}\sum_{i=1}^{100} sorted(x_i - \hat{x_i})^2 \,. \tag{5.2}$$

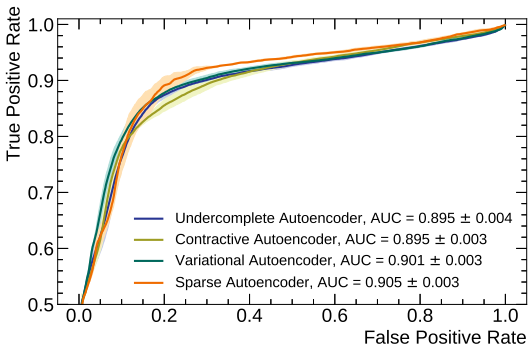


FIGURE 5.3: ROC and AUC of the autoencoder models using different regularization techniques. The bands correspond to variance computed after running the experiment five times using random weight initialization.

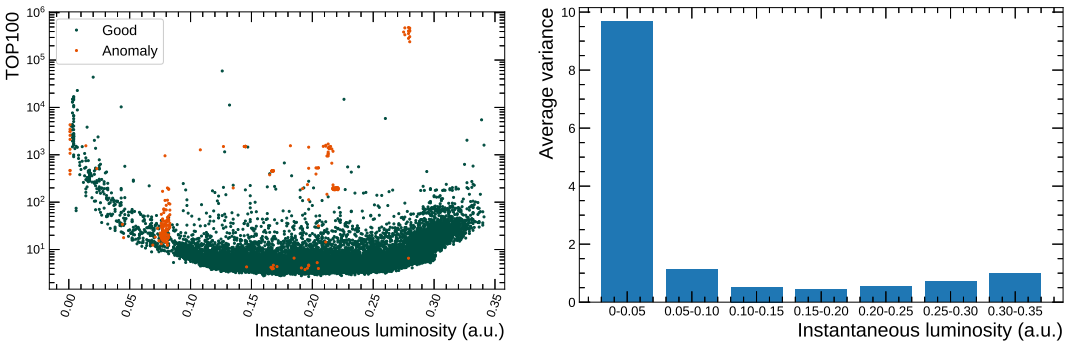


FIGURE 5.4: Anomaly score w.r.t. instantaneous luminosity (left) of sparse autoencoder and the average feature spread in different instantaneous luminosity regions (right).

5.2.2 Experimental Results and Discussion

The final ROC curves for models and their corresponding AUC are reported in Figure 5.3. All models show excellent performance, especially the sparse autoencoder. The two techniques of improving the results were investigated: sample weights and systematic feature selection. However, the overall performance gained from implementing any regularization is minimal, which calls for investigating more complex designs with the possible use of synthetic novelties as proposed in Kliger and Fleishman (2018).

Figure 5.4 shows the TOP100 error yield for each sample in the test set as a function of LHC instantaneous luminosity. The instantaneous luminosity is proportional to the number of collisions developing in each bunch crossing multiplied by the number of bunch crossings per second (see Section 2.2). The error is visibly higher in low and high luminosity regions. Hence the model is unable to capture full data variability. This dependence could also be caused by a smaller amount of samples coming from those regions. Sample weights were used in order to weight the error yield in those regions more, but no performance improvement was noticed. Adding additional autoencoder input carrying values of instantaneous luminosity has also been shown not to improve the performance.

Systematic feature selection was investigated to improve the classification results. Same features may harm the overall performance as they are close to being constant-valued and

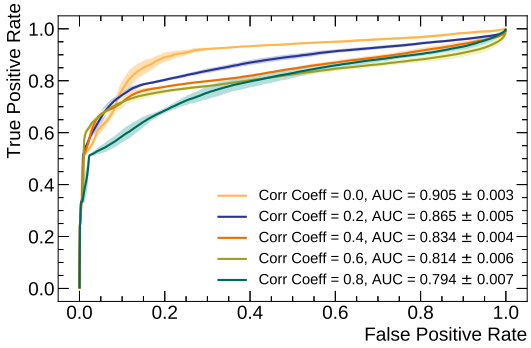


FIGURE 5.5: ROC curves for models using different input feature set. Feature selection does not improve overall ROC AUC but changes the shape of the curve. The bands correspond to variance computed after running the experiment five times using random weight initialization.

thus useless, or when it is impossible to reconstruct them accurately. Features with minimal ($\epsilon = 10^{-8}$) variability were removed. Subsequently, the Pearson Correlation Coefficient (PCC) ρ between training and reconstructed training data was calculated for each feature. Features below a specific threshold were removed from the data set, and the model was retrained. The table below summarizes the number of features corresponding to each condition:

Condition	Number of training features
-	2807
$\sigma > \epsilon$	2461
$\sigma > \epsilon$ AND $\rho > 0.2$	1518
$\sigma > \epsilon$ AND $\rho > 0.4$	1142
$\sigma > \epsilon$ AND $\rho > 0.6$	827
$\sigma > \epsilon$ AND $\rho > 0.8$	521

Figure 5.5 shows the performance of the sparse autoencoder trained with different preselections of the features. While the pruning seems to affect the AUC negatively, it shows minimal beneficial effects in maximizing the TPR under low FPR constraints.

5.2.3 Comparison with Supervised Anomaly Detection

As discussed before, the data of the CMS experiment undergo a slow but continuous evolution due to changing settings in the accelerator and detector operating conditions. In an evolving conditions context, the CMS Collaboration is looking for tools guaranteeing stable performance over time, even at the cost of slightly lower performance. The discrimination power of supervised ML algorithms is expected to evolve throughout the data acquisition campaign: the performance depends on the availability of data evaluated and lebelled by the experts, and thus available for training. Moreover, supervised methods are expected o have some intrinsic limit in detecting novel failure modes.

To test this hypothesis, the performance of XGBoost proposed in Chen and Guestrin (2016) (a supervised method previously suggested in Stankevicius et al. (2018) in the context of CMS DC), IF and the semi-supervised reference model, the sparse autoencoder are compared. The goal of the exercise is to compare the performance of the three methods emulating a realistic

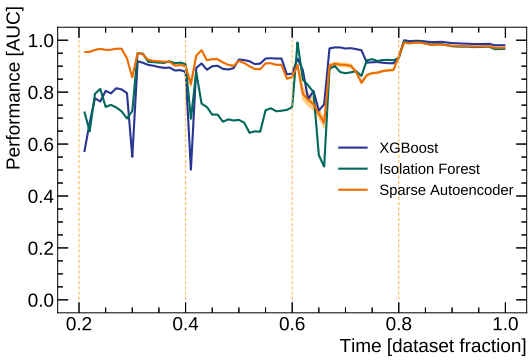


FIGURE 5.6: Performance of different strategies as a function of time. Compared are supervised AD XGBoost, semi-supervised AD sparse autoencoder and unsupervised IF. After every 0.2 fractions of the data set, each method is retrained on all of available past data points using ground truth labels. AUC is reported since last retraining.

deployment model. On this purpose, the algorithms are retrained on past data in chunks corresponding to about one month of data taking time (corresponding to about 20% of the data set).

Figure 5.6 shows the performance evolution for each model calculated since last retraining time. The visible performance drops, around 0.3, 0.4 and 0.65, are caused by the appearance of novel problems. The autoencoder performance is less affected by those events than the performance of XGBoost. Results show that the semi-supervised approach guarantees a more stable performance than supervised ones. Nevertheless, a fully supervised approach may still be a powerful addition to the proposed protocol, as its performance is frequently superior.

5.2.4 Interpretability of the Classification Results

Beyond high specificity and low FPR, a valuable model for the DC task needs to provide easily interpretable results allowing the certification experts to pinpoint the reason why a specific LS is lebelled as anomalous. In this respect, the autoencoder approach provides a clear advantage allowing to evaluate the contribution to the MSE of each input variable. Misbehaving variables can be easily singled out based on their high contribution to the overall error. Figure 5.7 illustrates one example of how this can be exploited on the CMS data. The features are grouped according to their sensitivity to a particular physics property. The plot of the absolute error allows the expert to identify the problematic area at a glance judging on the absolute size of the error for the variable or group of variables.

5.3 Conclusions and Practical Considerations

This work explored the usage of autoencoders for a semi-supervised novelty detection applied to the CMS physics data DC. The results presented in Section 5.2.2 proved that the autoencoders are robust to rare and newly emerging problems and may be successfully employed to tackle the novelty detection problem. However, unlike in Section 4.3, the different regularization techniques provide a minor increase in detection performance, suggesting

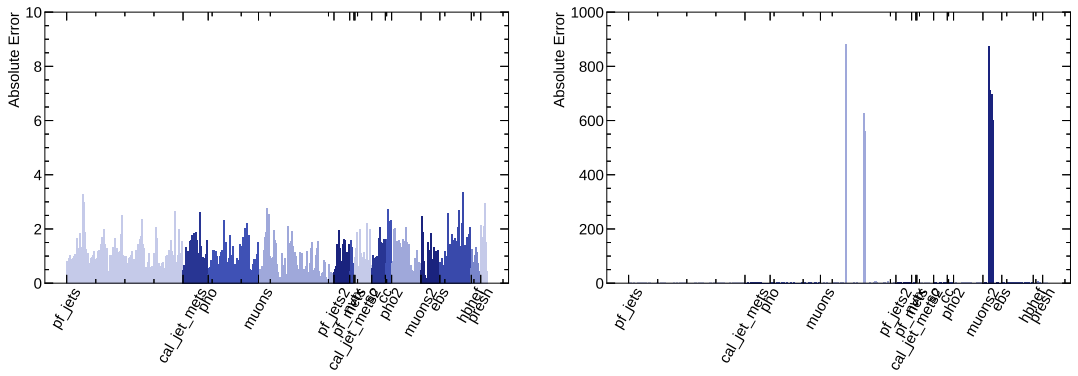


FIGURE 5.7: Reconstruction error of each feature for two samples. Different colours represent features linked to different physics objects. For a negative sample (left) similar autoencoder reconstruction errors are expected across all objects with small absolute scale. Anomalous samples (right) have visible peaks for problematic features (muons).

more complex solutions would be necessary in order to achieve better results. The results from Section 5.2.3 show superior performance of autoencoders on novelty detection tasks.

The proposed method monitors the distribution of several hundred physics quantities with very low time-granularity, allowing to identify emerging novelties promptly and to identify which ones among the input variables show an anomalous behaviour. This aspect of the interpretability of the results is a crucial feature for the physicists operating the tool. DC experts are validating the approach on more recently collected data from 2018 LHC run and coming from different PDs. Efforts to integrate the tool in the existing protocol as assistance for human experts were undertaken. Further studies are ongoing to consolidate a training and deployment strategy allowing the model to describe the evolving nature of the experiment data accurately. It is not necessary to classify all the data without any human intervention. Instead, the system can call for verification of questionable cases still limiting required human labour.

CHAPTER 6

Trigger Rate Anomaly Detection with Conditional Variational Autoencoders

The use case considered for this chapter is the monitoring of the CMS trigger system, which, as illustrated in Section 2.3, is the online data selection stage of all experiments at the LHC. The crucial role of the trigger system imposes stringent constraints on any AD methodology to be considered for production deployment: this includes the detection performance but also simplicity and robustness, for long-term maintainability.

Exploiting the rapid advancements in probabilistic inference, in particular, VAEs, for AD tasks remains an open research question in ML community. Relatively few works have been devoted to exploiting AD in modelling complex structured representations that effectively perform probabilistic inference (see Section 3.5.3). In most of them, the VAE architectures are specifically changed for specific sub-cases of AD with complex extensions. As discussed in Section 3.5, these contributions, e.g. Hendrycks et al. (2018); Wang et al. (2019), argue that training models only with inliers is insufficient and the VAE framework should be significantly modified to discriminate the anomalous instances. In this chapter, this idea is yet again challenged. The deep conditional generative architecture - CVAE - is exploited, together with a loss function and metric that targets hierarchically structured data AD. Hierarchical representation learning is a big challenge of DL, often obtained as a byproduct of data compression, as described in Bengio et al. (2013).

The focus of this chapter is on a semi-supervised AD, where the training set is generally free from outliers, and the examples of anomalous instances are not available. Moreover, the monitored system is a complex apparatus where only some of the parameters driving its behaviour are known *a priori* and measured. The observable x can be represented as a function of k (*known*) and u (*unknown*) vectors: $x = f(k, u)$. For a collection of samples, $x = [x_1, x_2, ..., x_n]$, some of the k parameters influence certain observable features, later called a *configuration group*. Features influenced by distinct k parameters are referred to as *uncorrelated features*. For visual interpretation, see synthetic data set description in Section 6.4. The monitoring algorithm needs to highlight instances where:

- a significant change on a single feature is observed, later call **type A** anomaly, *and*

- small but systematic changes on a group of features in the same configuration group (generated using the same k parameters) is observed, called **type B** anomaly.

On the contrary, samples with small statistical fluctuations on a group of uncorrelated features should not be considered problematic, as these are expected.

In summary, an algorithm needs to exploit known causal relations in data to spot both types of problems listed above. The algorithm needs to generalize to unseen cases and use data instead of relying on feature engineering. The algorithm has to be able to successfully disentangle k from u as underlying factors of variation of x to achieve the above-stated goals.

The structural modifications of the VAE architecture are introduced to cope with the requirements specified above. The resulting *AD-CVAE* model and corresponding loss function are described in Section 6.3. As the name suggests, the AD-CVAE specifically targets AD, as instead of a purely generative model. As such, it introduces conditioning through limited modifications to the vanilla VAE architecture, which is consistent with the simplicity goals.

The versatility and effectiveness of the method are initially demonstrated on data sets commonly used for benchmarking ML applications. A synthetic data set is then used to demonstrate the applicability to the CMS TRM use case. These data sets are described in Section 6.4. The experimental results, presented in Section 6.5, show that the AD-CVAE provides excellent performance for the two types of anomalies described above.

Overall, the main contributions of this work are as follows:

- an anomaly metric associated with CVAE architecture is used, providing superior performance on both types of target anomalies and both classical ML and physics specific data sets (Section 6.3.3);
- a loss function that allows the model to learn the optimal reconstruction resolution is used (Section 6.3.1); *and*
- alternative experimental setup for AD on the Modified National Institute of Standards and Technology (MNIST) data set is proposed (Section 6.4.2).

6.1 Motivation: Monitoring of the CMS Trigger Rate

This section describes hierarchical nature of the CMS trigger system, which is essential to understand before building robust monitoring strategy. The overview of the CMS TRM application is available in Section 2.4.4.

The trigger system regulates the massive data deluge coming from the observed collisions: its role is critical since any problem could result in a severe data loss. For this reason, besides the sub-detectors providing input to the trigger system, also the output rates need continuous monitoring. The event acceptance rate can be affected by several issues, e.g. detector malfunctions or network and software problems. Depending on the origin of the problem, the rate of a specific trigger node could change to unacceptable levels (critical cases include dropping to zero or increasing to extreme values). The goal of the TRM is to alert the shift crew in case suspicious rates are measured, calling for problem diagnosis and intervention.

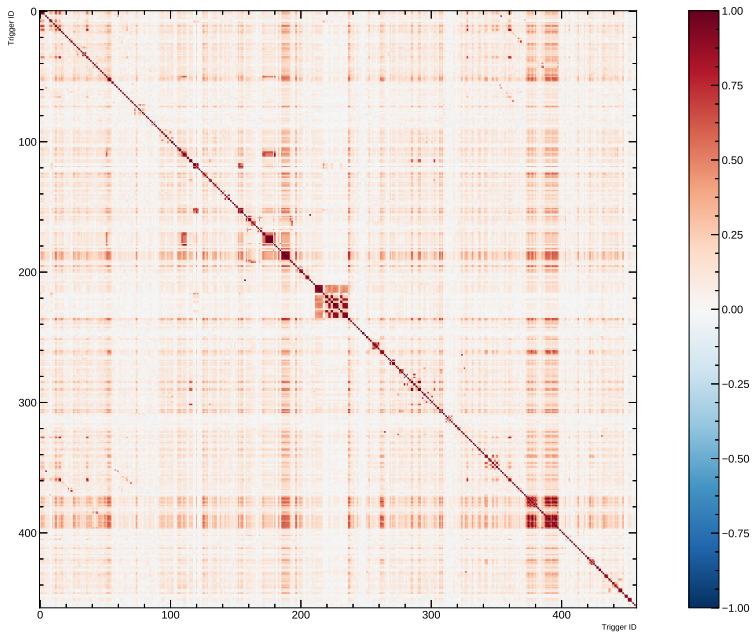


FIGURE 6.1: Correlations between 458 HLT rates of LHC fill 6291 of Run 2. The calculation is performed after correction for down-trend of all trigger rates using first differences.

As explained in Section 2.4.4, the current production TRM system is based on the comparison between the observed per-node rate and its reference value for the measured PU value. These references are derived from fitting previously collected data with analytical models.

While the current implementation is quite effective in spotting erratic changes for a single node, it is less sensitive to collective changes on several nodes that could equally affect the overall acceptance rate. In particular, about 600 nodes of the HLT can be grouped in several configuration groups, showing strong correlations in their acceptance rate variations, as shown in Figure 6.1. The underlying process of the different configuration groups can be of different origin. It can be driven by the physics processes when different nodes select the same physics objects with different requirements, e.g. different requests on its energy. Alternatively, it can be related to the utilization of the same sub-detector component or software component across different nodes. The most critical and easily measurable origin of these correlations is, however, related to the fact that different HLT nodes can be fed by the same selection of L1 Trigger nodes. Each HLT node has a direct, pre-configured link to a set of L1 Trigger nodes through a specific configuration. The connection between L1 Trigger and HLT nodes can be seen as a hierarchical directed graph connecting L1 Trigger to HLT nodes, as schematically shown in Figure 6.2. The configuration changes infrequently, i.e. nodes are added, disabled, or corrected.

As a result, when building an extended TRM application, the performance of the HLT system can be modelled as a function of the L1 Trigger input rates. While these rates are measured and available, all the other potential sources of correlation are not easily estimated and determinable *a priori* due to the complexity of the system. For this reason, some *unknown* factors must be introduced that are left as additional degrees of freedom of the model. Extended TRM needs to be aware of the existence of configurations groups based on these two classes of inputs. The extension needs to:

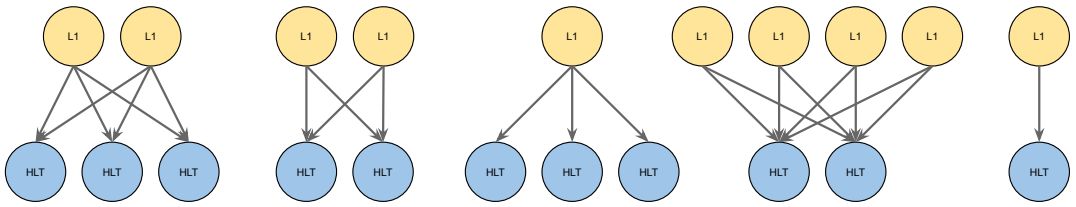


FIGURE 6.2: Simplified, schematic graph inspired by the trigger system configuration. Blue nodes represent HLT while yellow L1 Trigger. Each link is unidirectional starting from yellow nodes. For each LHC fill the graph has a few hundred nodes spread approximately equally between HLT and L1 Trigger. The connection between L1 Trigger and HLT nodes can be seen as a hierarchical directed graph from the L1 Trigger to the HLT system.

- single out isolated problems on individual HLT nodes (to reproduce the functionality of the current TRM); *and*
- highlight problems present across similar HLT nodes (novel strategy).

At the same time, the new algorithm needs to be resilient against statistical fluctuations in the rate of a single HLT node, thus limiting the number of FPs.

The anomaly ground truth is unavailable, and a supervised AD strategy is not applicable under given settings. The available quality flags do not necessarily correspond to failures in the trigger system. For instance, a flag marking a failure in the muon system does not imply a noticeable drop in all muon related trigger nodes as the failure may be related in only one out of four (see Section 2.3.2) muon subsystems. Besides, a failure derived from monitoring the detector components such as the occupancy plots (see Chapter 4) may not be evident from the trigger rate data.

The inference has to be run on all trigger nodes every LS, a time interval of about 23 s, which does not impose any challenging constraint in terms of timing.

6.2 Conditioning on Observed Factors of Variation

As pointed out in the previous section, the HLT rate depends on a set of known and unknown factors. Because of the nature of the target application, the algorithm has to be conditional to correctly model the system. In layman terms, the structure of the data is partially known, as the HLT rates are associated with available L1 Trigger rates.

This raises the general question of disentanglement (see Section 3.5.2). To correctly model the trigger system, the algorithm has to successfully disentangle the HLT rate dependence on L1 Trigger rate and all other unknown processes. In light of the results of Locatello et al. (2019), and in particular Theorem 1, the disentanglement objective in generative models, cannot be met by the fully unsupervised VAE architectures. An alternative is to leverage the disentanglement through the known information using structured architectures based on the CVAE.

In broad terms, the CVAE is a conditional directed graphical model where input observations modulate the prior on latent variables that generate the outputs. Figure 6.3 shows an

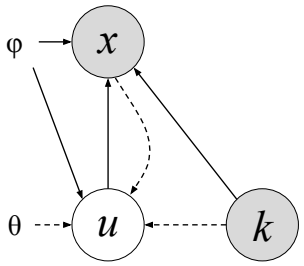


FIGURE 6.3: An example of CVAE as a directed graph. Solid lines denote the generative model $p_\theta(x|u,k)p_\theta(u)$. Dashed lines denote variational approximation $q_\phi(u|x,k)$. Both variational parameters θ and generative parameters ϕ are learned jointly.

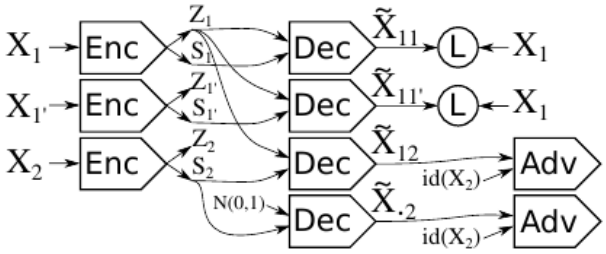


FIGURE 6.4: Architecture of CVAE based on Mathieu et al. (2016) proposal. The inputs x_1 and x_1' are two different samples with the same label, whereas x_2 can have any label.

example of CVAE. CVAE models the distribution of high-dimensional output space as a generative model conditioned on the input observation. In the target use case, x corresponds to the HLT rates that are conditioned on k, which is a vector of L1 Trigger rates and additional, unknown phenomena u. The use of the underlying VAE framework guarantees that the method generalizes well to unseen observations. In recent years several CVAE architectures have been proposed. The literature usually approaches the CVAE with a semi-supervised loss as the model has to leverage a mix of labelled and unlabeled data as it has been previously done by, e.g. Kingma et al. (2014); Siddharth et al. (2017). Below, an overview of notable works is presented, while the relation to the proposed AD-CVAE model is given in Section 6.3.4.

Kingma et al. (2014) proposed a new architecture based on VAE framework, named M1+M2. The underlying representation is divided into z and y part where y corresponds to label information for a subset of observed data x. The training is done in two concurrent parts called the M1 and M2. For annotated samples, the inference model is conditioned on y, $q_\phi(z|y,x)$ (M1). For unlabelled observations, the inference is denoted as $q_\phi(z,y|x)$ (M2). The architecture targets classification as well as generation. It is trained in a semi-supervised approach using both labelled and unlabelled data, which contributions are weighted in the loss function.

Mathieu et al. (2016) defined a conditional model focused on generative usage. It presents a thorough discussion of the challenges facing the enforcement of disentangled representation in the conditional VAE approach. Their probabilistic model is $p(x|z,s)$, where variables s and z denote the *specified* s and *unspecified* (analogous to known and unknown) factors of variation respectively. Specifically, in the paper, s is a continuous version of a class label to enhance the generative capabilities of the model. Both s and z are assumed to be marginally

independent to promote disentanglement at a semantic level. As usual, the likelihood function $p_\theta(x|z,s)$ is a decoder network. The approximate posterior is modelled using an independent Gaussian distribution, $q_\phi(z|x,s) = \mathcal{N}(\mu, \sigma I)$, whose parameters are specified via an encoder network, and a deterministic function $\hat{s} = f(x)$, which together form the encoder. Figure 6.4 presents this setup.

The critical observation made by Mathieu et al. (2016), is that the model cannot be trained by minimizing the log-likelihood alone. Nothing prevents all of the information about the observation to be captured by the z component. The decoder could learn to ignore $\hat{s}$, while the approximate posterior could map data generated from the same class to different regions in the z space. The architecture has to be augmented with adversarial training that swap observations within a class and the loss function must include the discriminator loss.

CVAE from Sohn et al. (2015) builds upon the VAE framework for structured output prediction. The model does not have to leverage unlabeled data. As such the training is performed in a supervised manner (as the corresponding labels are always available) which explicitly disentangle factors k and u. Similarly to Mathieu et al. (2016), the probabilistic model $p_\theta(y|z,x)$ is trained jointly with the encoding model $q_\phi(z|x,y)$.

CVAE are also related to the ICA model proposed in Hyvärinen et al. (2019). In order to cope with the identifiability result of Hyvärinen and Pajunen (1999), the model assumes that each component s_i is statistically dependent on some *fully-observed* multidimensional random variable u, but conditionally independent of the other components:

$$p(s|u) = \prod q(s_i, u) \, . \tag{6.1}$$

For this setting, effective identifiability results are derived from a method very close to the one of Sohn et al. (2015): a deep conditional generative model for structured output prediction, trained with full supervision.

6.3 Training CVAEs for AD

To target the problem presented in Section 6.1, an architecture, later called AD-CVAE, is proposed together with the corresponding loss function for optimizing the model parameters and anomaly detection metrics.

6.3.1 The Optimal Reconstruction Resolution

The original works on VAEs by Rezende (2014); Kingma and Welling (2013) proposed a full (diagonal) Gaussian observation model, that is

$$P_\theta(x|z) = \mathcal{N}(\mu, \sigma I) \, . \tag{6.2}$$

In this model, both the multidimensional mean vector and the multidimensional variance vector are to be learnt. For reasons that are discussed in Section 6.5, this comprehensive approach has often been much restricted. In practical applications the VAEs evaluate the reconstruction loss as an MSE between the data x and the output of the decoder. Such an

approach suffers from a very serious issue. It is equivalent to setting the observation model $p_\theta(x|z)$ as a normal distribution of fixed variance $\sigma = 1$. Indeed, the log-likelihood of a normal distribution with a fixed variance of 1 is given as

$$-\log \mathcal{N}(x; \mu, 1) = \|x - \mu\|^2 + \log(\sqrt{2\pi}) \,. \tag{6.3}$$

Fixing the variance this way can be detrimental to learning as it puts a limit on the accessible resolution for the decoder. The generative model has a fixed noise variance on its output, making it impossible for it to accurately model patterns with a characteristic amplitude smaller than that. However, unless *a priori* knowledge suggests it, there is no guarantee that all patterns of interest would have such a sizable characteristic amplitude. The model can learn the variance of the output of the decoder feature-wise (i running as the dimensionality of the data vectors x):

$$-\log p_\theta(x|z) = \sum_i \frac{(x_i - \mu_i)^2}{2\sigma_i^2} + \log\left(\sqrt{2\pi}\sigma_i\right) \,. \tag{6.4}$$

Learning the reconstruction variance allows the model to find the **optimal reconstruction resolution** for each feature of the data, separating the intrinsic noise from the actual data structure. However, as discussed in Lucas et al. (2019); Zhao et al. (2017) this approach poses additional challenges on the optimization process, which are going to be discussed in Section 6.5.

6.3.2 The Structure and the Loss Function

In the proposed setup, there are three types of variables, see Figure 6.3: for random observable variables x, u and k are *marginally independent variables*. If one assumes the conditional dependence between u and k, then $u = f(k) + \epsilon$, where ϵ represents noise independent from k. The likelihood function $p_\theta(x|u,k)$ is then equivalent to

$$p_\theta(x|f(k),k,\epsilon) = p_\theta(x|k,\epsilon) \,. \tag{6.5}$$

There is no predefined semantics on latent variable u as a generative model is not the interest *per se*. Instead, u should carry minimum information necessary to generate x. In the above case, ϵ carries such information, and we are back to marginal independence.

The variable u allows for modelling multiple modes in the conditional distribution $p(x|k)$ making the model sufficient for modelling one-to-many mapping. The conditional likelihood function $p_\theta(x|u,k)$ is formed by a non-linear transformation, with parameters θ.

ϕ is another non-linear function that approximates inference posterior $q_\phi(u|k,x) = N(\mu, \sigma I)$. The ϕ and θ are implemented as DNNs with non-linear activation functions. The schema of the network architecture, corresponding to a graph from Figure 6.3 is shown in Figure 6.5. This setup allows for fast detection using stochastic feed-forward inference.

The architecture from Figure 6.5 is trained efficiently in the framework of stochastic gradient variational Bayes. Recall the VAE training objective from equation 3.33. In CVAE case, to

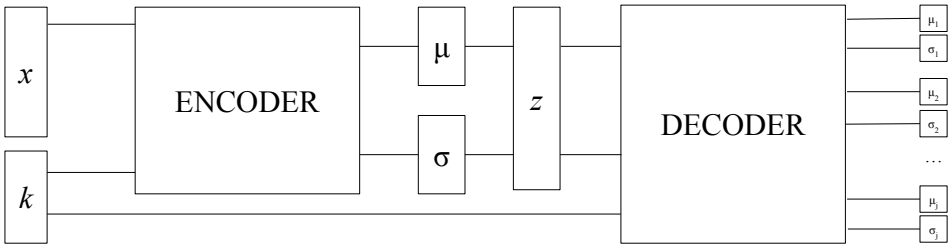


FIGURE 6.5: Architecture of AD-CVAE targeting structured AD based on the VAE framework. The output μ and σ correspond to optimal reconstruction resolution. Observable data x depends on z (capturing non-observable factors of variation u) and k vectors.

approximate ϕ and θ, the following ELBO is optimized:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|k,x)}[\log p_\theta(x|z)p_\theta(x|k)] - \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x,k)||p(z)) \,, \tag{6.6}$$

where z (Gaussian latent variable) intends to capture non-observable factors of variation u.

After inserting equation 6.4 as the reconstruction objective to the loss based on the ELBO term from equation 6.6, the final loss of AD-CVAE is

$$\mathcal{L}_{\text{AD-CVAE}}(x,k,\theta,\phi) = \sum_i \frac{(x_i-\mu_i)^2}{2\sigma_i^2} + \log\left(\sqrt{2\pi}\sigma_i\right) + \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x,k)||p(z)) \,. \tag{6.7}$$

6.3.3 The Metric

The final and essential question is how to use the trained AD-CVAE model in order to address the target problem: build a robust AD mechanism. As explained in Section 6.1, there are two failure scenarios: type A for significant errors on isolated features and type B for shallow but correlated ones. Conveniently, the loss function from equation 6.7 can be broken up into two components to target these two separate scenarios. A final binary label is needed and easily derived from these metrics. Moreover, the separability of the metrics provides additional information, that allows for differentiating between the failure modes. Here, the two metrics are first defined, and the discussion and motivation follow.

Once the model parameters are learned, one can detect anomalies:

- of type A with average infinity norm of the reconstruction loss $m_A = ||\frac{1}{\sigma}(x-\hat{x})^2||_\infty$, where $\hat{x}$ is the reconstructed mean and σ is the reconstructed variance of decoder output; *and*
- of type B with KL divergence $m_B = \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x,k)||p(z))$, see equation 3.38, known as information gain.

In the first case, an anomaly is identified on a single feature. For a given data point (x,k), the evaluation of the loss of the VAE at this data point $\mathcal{L}(x,k)$ is an upper-bound approximation of $-\log p_\theta(x|k)$, measuring how unlikely the observation x is to the model given k. AD-CVAE thus provides here a model that naturally estimates how anomalous x is given k, rather than how anomalous the couple (x,k) is. That means that a rare value of k associated with a proper value for x should be treated as non-anomalous, which is the goal. The

binary indicator is obtained by thresholding the value, a typical strategy for the AD. With thresholding, the choice of the infinity norm of the reconstruction error instead of the mean is required. A mean of the reconstruction error would be uninformative when most of the features do not manifest abnormalities and, as a consequence, lower overall anomaly score.

As argued in Gemici et al. (2017), the $\mathbb{D}_{\mathrm{KL}}$ measures the amount of additional information needed to represent the posterior distribution given the prior over the latent variable being explored to explain the current observation. The lower the absolute value of $\mathbb{D}_{\mathrm{KL}}$, the more predictable state is observed. The $\mathbb{D}_{\mathrm{KL}}$ was then used as a *surprise* quantifier, e.g. in Gemici et al. (2017); Eslami et al. (2018) when model was exposed to held-out images. Nalisnick et al. (2019); Snoek et al. (2019) explored $\mathbb{D}_{\mathrm{KL}}$ as an indicator of OOD samples.

For the type B outliers, the expected anomaly reinforces patterns in data in a systematic way. It is then expected that not calibrated model allocates such information using the the latent bits, allowing for a successful reconstruction. On the other hand, changes on uncorrelated features will be removed in the encoding process, resulting in low reconstruction likelihood. Hence anomalous input yields higher values of m_B and likelihood at the same time. Thus, m_B must be detached from the reconstruction part of the loss function as combining metrics is detrimental to the detection results.

Because of two separate failure scenarios, the metrics are not combined in one overall score but rather use logical `OR` to determine anomalous instances.

6.3.4 Discussion and Related Work

The CVAEs were successfully used for AD in the multivariate time series data Suh et al. (2016), intrusion detection tasks Lopez-Martin et al. (2017), and robot time series data Sölch et al. (2016). However, the past contributions tackled very different AD problems to the one discussed in this chapter and used much different structural approach. The authors of Suh et al. (2016) approach did not use $\mathbb{D}_{\mathrm{KL}}$ as the metric for the anomaly score. Instead, just the negative log-likelihood of the corresponding ELBO objective was chosen. Lopez-Martin et al. (2017) neither uses $\mathbb{D}_{\mathrm{KL}}$ for anomaly score. Instead, a forward pass with changing deterministic input to the decoder is used to generate predictions. Sölch et al. (2016) based their architecture on the RNN and among several anomaly scores, they based the results on the full lower bound, while $\mathbb{D}_{\mathrm{KL}}$ was not analysed in isolation.

The AD-CVAE model is inspired by the contributions listed in Section 6.2, but it focuses specifically on AD tasks, as the generation of realistic images nor leveraging unlabelled instances during training is not the focus. The setup is insufficient for generative purposes. As shown in Mathieu et al. (2016), a supplementary adversarial system is needed for such an objective. However, the AD task is more straightforward as it is not necessary to generate realistic (however they still should be accurate) outputs of the generator. Opposite to Mathieu et al. (2016), in the context of this work, the adversarial system was not necessary to prevent the network from ignoring k. Besides, this kind of proposed regularization will not help with a training set that is contaminated with outliers (which may be the case, e.g. Section 6.4.2). Using complex setup can give the encoder possibility to store too much information about the anomalies in z and harm both types of detection metrics. The AD-CVAE is similar to M2 model from Kingma et al. (2014) but is simplified for the targeted AD needs.

Strictly speaking, the approach should not be called a semi-supervised one, as it does not refer to training CVAE with unlabelled data as in the cases described in Section 6.2. A more proper name is instead a partially supervised AD strategy: using primarily inliers (as in class-modelling or OCC) for training and making use of the observable k.

Finally, balancing the ELBO with the β parameter (see Section 3.5.2) became a *de facto* standard procedure when training the VAE, even when disentanglement is not directly an objective of a given study. Despite providing good empirical results, such protocol creates a need for fine-tuning yet another additional hyper-parameter. In the experiments it is found that the optimal reconstruction resolution empirically gives similar results to associating a fine-tuned β, while removing the need to tune said hyper-parameter.

6.4 Experimental Setup

This section will first provide an overview of the experimental approach. Then, it will present in detail the benchmarks used for the experiments. The benchmarks are *specially built* with the target data and detection objectives in mind. All the experimental results are in Section 6.5.

6.4.1 Overview

Relevant benchmarks had to be defined for the approach. From a methodological point of view, it is necessary to experiment on more than the trigger behaviour. Although these are the real-world data, they do not introduce specificities of other applications, e.g. AD on images. A more comprehensive range of use cases is needed. Firstly in order to check the effectiveness of the AD-CVAE on out-of-scope applications. Secondly, to have access to a parametrizable fault behaviour, in order to explore in detail the performance of the strategy. Two very different types of data are used to achieve these goals. For the first goal, the MNIST and its variants are used. For the second goal, a Synthetic Trigger (STRI) is defined, a simplified and modular model of the real data, as a GMM. As the experiments in the next section will show, it turns out that STRI is a credible proxy for the trigger data. Of course, these experiments are followed by the experiments on the target data.

The associated preprocessing is described in this section. As such, the data sets do not suffice to make a benchmark. They must be enriched: for instance, MNIST does not offer a clear-cut notion of anomaly. On the other hand, training and evaluation demand it. Concerning training, as AD-CVAE is partially supervised (as defined in Section 6.3.4), uncontaminated train sets are required. Therefore, we have to define the normalcy concept; as self-definition always runs the risk of experimental bias, we took special care of creating a *parameterized truth* (a completer). In this sense, this section proposes two new AD benchmarks, one based on MNIST and the other on GMM as a case for structured data. Because most procedures are specific of the benchmarks, the rest of the section is organized along with their descriptions.

6.4.2 MNIST and Fashion-MNIST Benchmark

Data Set

The performance of the proposed method will be assessed using the MNIST, and the more recent and more challenging Fashion-MNIST data sets, introduced in LeCun et al. (1998) and Xiao et al. (2017) respectively. The data sets contain grey-level images of handwritten digits or pieces of clothing. Although the target application deals with numerical data, not images, this preliminary experiment serves as an example of possible out-of-scope, real-world applications.

Structure

Handwritten digits in MNIST data set belong to a manifold of dimension much smaller than the dimension of x (28x28 pixels) because the majority of random arrangements of pixel intensities do not look like handwritten digits. Intuitively, this dimension should be at least the size of 10 as the number of classes suggests. Nevertheless, larger latent space needs to be accommodated as each digit can be written in a different style. Similar intuition applies to Fashion-MNIST as this data set also has ten target classes of clothing types, but there is variability inside a class, e.g. a type of shoe.

The class and style distinction has been abundantly exploited in the disentangling context Mathieu et al. (2016); Kingma et al. (2014). Importantly, a human observer could regard digits as anomalous because of the latent features capturing handwriting styles (irrespective and not captured by the class label), when they describe original, unconventional handwriting styles. For instance digit 4 with style resembling digit 9, or extremely rotated digit could be considered as anomalous.

In the experimental setup, a class label is assigned to vector k while u should accommodate information about other latent features of x including style. The x are the input pixel values. Throughout the experiments, the original train-test splits are used with 10000 test samples.

Tagging Anomaly

Past works on AD with MNIST data set arbitrarily assigned one of the classes as anomalous. For instance, digit 0 was considered abnormal, while other digits were considered as inliers. Although this approach is straightforward, its methodological quality is challenged here.

In the following, a simple alternative method for evaluating AD in the MNIST context is defined based on two ideas. Firstly, the class is irrelevant, and the character style is the key; secondly, as entirely objective tagging of those kinds of anomalies is impossible, and a fully configurable tagging oracle is needed. The two are linked by the fact that the configuration parameter will capture the capacity to accept more or less bizarre writing style.

The overall evaluation procedure is described in algorithm 2. The oracle is obtained by a traditional multiclass classifier $\mathcal{M}$. $\mathcal{M}$ is trained in `Oracle` procedure in a supervised manner on a fixed training set. On the test set, the classification score s, equivalent to classification error $1 - p(k|x)$, is used to tag each sample, as an outlier or inlier. Intuitively, the higher the s, the more anomalous the data point. s is continues and in $(0, 1)$ range.

The purpose of the `Detection` procedure is to provide performance indicators that do not depend on any specific and subjective choice of anomalies. As this is not possible with a single figure, the output of the `Detection` procedure is a discretized version of a function.

Given an AD algorithm A trained on the fixed training set, and a test set, the `Detection` first computes the anomaly scores. In the second step, a set of true labels is selected based on the previously obtained classification score s and a chosen strictness τ. For a given τ, different truth labels are chosen for the test data. Then the `Detection` returns an AUC as a function of varying strictness τ. For instance, the number of anomalous samples reported for $s > \tau, \tau = 0.99$ is much smaller than the ones reported for $\tau = 0.01$. Naturally, the second set of anomalies will contain all the instances from the first. With decreasing τ, more test samples will flip to anomalous. The strictness τ is a proxy of how strict the oracle is in its decision. By reporting only the results for an arbitrary value of τ, one may miss regions where a given algorithm performs worse than the baseline. Instead, plotting ROC AUC for multiple τ values, $\tau \in (0, 1)$ with small increments of 0.01 gives an objective assessment of the algorithm's performance. The same procedure is applied for Fashion-MNIST data set.

In this experiment, the oracle is based on the LeNet-5 classifier proposed in LeCun et al. (1998) and minimizing the cross-entropy loss.

Algorithm 2 Proposed AD Evaluation Strategy on MNIST and Fashion-MNIST data sets

procedure ORACLE(Model $\mathcal{M}$, Data X_{train}, Data X_{test})
  $\mathcal{M} \leftarrow X_{train}$ ▷ Training classifier with $-\sum_i k_i \log(\mathcal{M}(x_i))$ loss
  $s = \mathcal{M}(X_{test})$ ▷ Evaluate log loss
  return s
procedure DETECTION(Algo A, Data X_{test}, Outlierness s)
  scores $= A(X_{test})$ ▷ Get anomaly score
  $t = 0.01$, $p = []$
  while $t < 1$ **do**
    truth $= s > t$ ▷ Get binary labels
    p.insert(AUC(truth, scores)) ▷ Get ROC AUC for τ
    $t = t + 0.01$
  return p

Alternatively, the performance of AD algorithms can be subjectively assessed using test data by directly reporting instances regarded as most anomalous.

6.4.3 The COIL-100 Benchmark

Data set

The Columbia Object Image Library (COIL)-100 data set, described in Nene et al. (1996), contains 7200 images of 100 different classes of objects. Each class contains 72 pictures taken at pose intervals of 5°. The original images are downsized to a size of 32×32 pixels.

Structure

Similarly to MNIST and Fashion-MNIST experiments, for the COIL-100 experiments, the k is assigned a class label. In this experiment, the cases of mislabelling are targeted, i.e. associating incorrect values of x to k. The proposed CVAE-based architecture is evaluated on a

test composed of samples drawn from $c \in [10, 15, 20, 25, 30, 40]$ random classes and extended with 20% of outliers. Those anomalous test samples, drawn from randomly selected non-training classes, are assigned a random training class label. In other words, the experiment is merely an example of mislabelling detection. The model is trained and validated on 80% and tested on 20% of selected training classes.

6.4.4 STRI Benchmark

Data set

The STRI data set uses normally distributed ($\mu = 0$, $\sigma = 1$), continuous and independent latent variables u and k. Observable x is a product of u, k and additional noise ϵ with given configuration constraints

$$x_j = f_j(\vec{u}) \cdot \sum_{i=0}^{m} \mathbf{S}_{ji} k_i + \epsilon \,, \tag{6.8}$$

where j is a feature index for $\vec{x}$ in $\mathbb{R}^n$. A binary matrix $\mathbf{S}$ describes which k is used to compute feature j:

$$\mathbf{S} = \begin{array}{c} \\ x_0 \\ x_1 \\ \vdots \\ x_n \end{array} \begin{array}{c} \begin{array}{cccc} k_0 & k_1 & \cdots & k_m \end{array} \\ \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \end{array}. \tag{6.9}$$

Finally, the function $f(\vec{u})$ describes which u enters the product that defines each feature j:

$$f_j(\vec{u}) = \prod_o u_o \,. \tag{6.10}$$

S and $f(\vec{u})$ stay unchanged across each sample in the data set, but the values of k and u do change. For simplicity, each feature j depends only on one k and the dependence is equally distributed. For instance, the first column x_0 can use k_0, u_1 and u_4: $x_0 = k_0 u_1 u_4$, x_{99} may be generated using k_4 and u_0. Finally, the values of o and m can be changed, effectively resizing k and u.

Structure

The synthetic data set is a version of GMM that could be solved using the EM algorithm as described in Section 3.3. However, it is just implemented as an initial benchmark that proxies the trigger data set.

For majority of tests, the samples are generated with x being 100-dimensional ($n = 100$) and $m = o = 5$. However, further experiments evaluate how changing values of m and o affect the final results. An example of a correlation matrix between features in such data set can be seen in Figure 6.6, which is a proxy of correlations between trigger nodes, as seen in Figure 6.1.

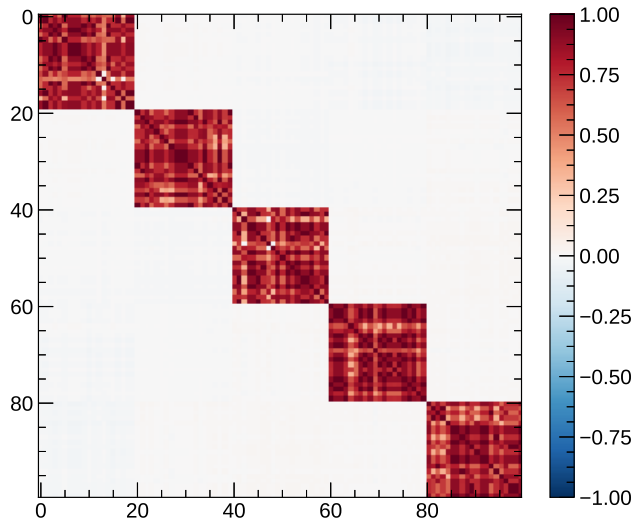


FIGURE 6.6: Correlations between features in the synthetic data set for $m = o = 5$ and $n = 100$. The data set was prepared as a proxy of the trigger system behaviour, seen in Figure 6.1.

Test set	**Description**
Type A Inlier	Generated in the same process as training data
Type A Anomaly	$s\sigma$ change on ϵ for a random feature
Type B Inlier	$s\sigma$ change on ϵ for a random set of $\frac{n}{m}$ features
Type B Anomaly	$s\sigma$ change on ϵ for a random configuration group

TABLE 6.1: Types of test data for the synthetic and trigger rates data set

Test samples are generated according to Table 6.1. The lower the value of s the more difficult an anomaly is to detect. In Gaussian distributions samples laying at 1σ are not considered as outliers. However, the proposed artificial change disturbs the natural relations between features. Hence, those changes should still be highlighted as anomalous as they do not follow the defined generative process.

The synthetic data set has 10000 training samples, and 4000 test samples (1000 inliers and 1000 outliers for each problem) are generated.

6.4.5 Trigger Rates Benchmark

Data set

The prototype implementation uses four L1 Trigger nodes that seed six distinct HLT nodes each ($6 \times 4 = 24$ in total). The rates from LHC Run 2 are extracted only from samples where all chosen nodes are present in the configuration. Each sample corresponds to one LS. In total, there are 102895 samples at hand, which are then split into training, validation, and test sets. The test set contains 2800 samples, which corresponds to one, last available LHC fill.

In the early phases of this study, it was noticed that the currently used analytic models (see details in Section 2.4.4) are fitting the HLT rate dependence on PU accurately, unless obvious modelling mistakes are made, see Figure 2.12. The fits generated by the TRM software thus

can be used as a preprocessing step in order to normalize the data and solve the problem at hand independently from PU variability. The recorded rate R is corrected by the predicted rate $\hat{R}$ to remove the general downtrend of recorded rates and their higher dependence on average PU. Obtained values are divided by TRM generated sigma, $\sigma_{R-\hat{R}}$, reported for each node individually. The final data point is obtained as

$$x = \frac{R - \hat{R}}{\sigma_{R-\hat{R}}} \,. \tag{6.11}$$

In this setup, the data across different PU ranges and different nodes are normalized. Final distributions have fixed variance $\sigma = 1$ and are centred with a mean close to $\mu = 0$.

For the experiment targeting the specific case of CMS TRM, x is assigned the HLT rate and k the L1 Trigger rate values.

Anomaly Tagging

As pointed out in Section 6.1, operators set quality labels for each CMS sub-detector and each LS. A contribution from all subsystems composes the global quality flag, and LS could be regarded as bad due to under-performance of a detector component not related to the set of trigger nodes chosen or not related to the problem to solve. Hence those labels cannot be used in the test set. Instead, hypothetical situations are considered. They are likely to happen in the production environment, similar to those used for the synthetic data set. Two groups of test data sets are generated by manipulating the test samples, similarly to the synthetic data set (see Table 6.1). Isolated problems on one of the HLT nodes (type A), and problems present across HLT nodes seeding the same L1 Trigger node (type B) need to be detected.

6.5 Experimental Results

The models are trained using Keras Chollet et al. (2015) with TensorFlow Abadi et al. (2016) as a backend using Adam Kingma and Ba (2014) optimizer and with early stopping Vincent et al. (2010) criterion. μ-SVM and IF serve as baselines for the first two experiments. Each model uses a concatenated class label to the input vector for a fair comparison. For the MNIST, Fashion-MNIST and COIL-100 experiment, the negative log-likelihood from equation 6.6 is estimated by cross-entropy error. After showing the effectiveness of the proposed algorithm in comparison to the vanilla VAE on standard ML data sets, the optimal reconstruction resolution proposal is examined on synthetic and physics data.

MNIST and Fashion-MNIST Benchmarks The most anomalous samples in the MNIST and Fashion-MNIST test set for each AD method are shown in Figure 6.7. Author's subjective eye regards the samples yield by β-CVAE as the oddest. The μ-SVM selects only one class as anomalous, while IF yields bold samples as problematic.

Figure 6.8 shows a more objective assessment. For changing values of classification error strictness τ, the ROC AUC is reported for the baseline AD algorithms and a vanilla VAE. It is noticeable that for vanilla VAE the $\mathbb{D}_{\mathrm{KL}}$ is not a useful anomaly indicator at all, as the information in the latent layer is expected to be entangled and mostly dominated by

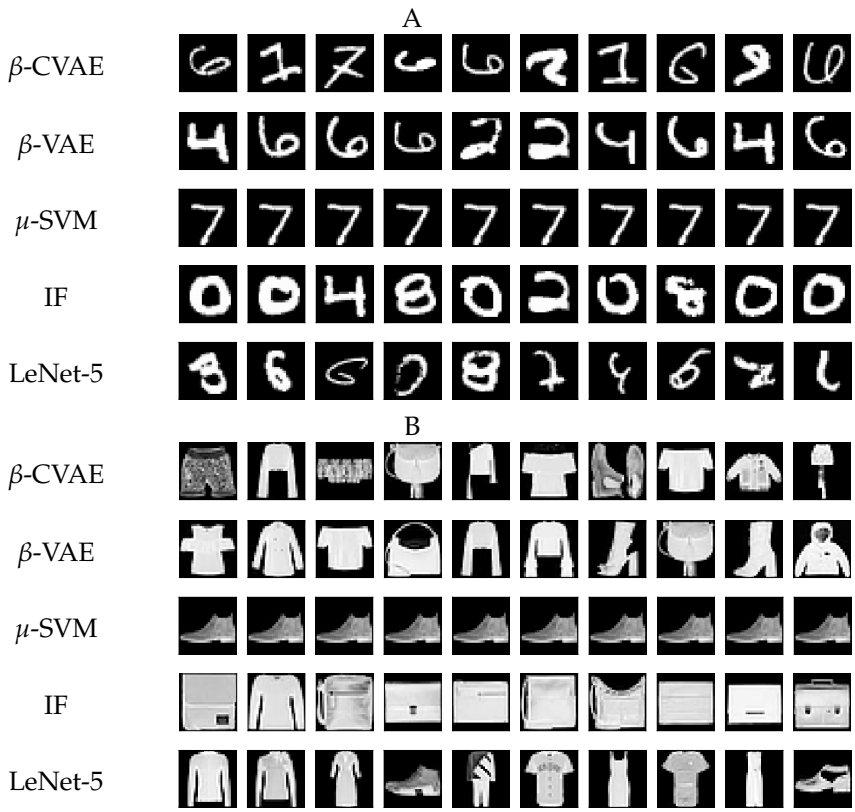


FIGURE 6.7: Most anomalous samples in the test set for MNIST (A) and Fashion-MNIST (B) data sets for each AD method. The samples reported for LeNet-5 classifier are the ones having the biggest classification error.

the class-label value. Changing architecture to β-CVAE ($\beta = 0.1$) turns $\mathbb{D}_{\mathrm{KL}}$ to a powerful anomaly indicator, which outperforms other baseline techniques. Both IF and VAE oscillate around 0.5 ROC AUC.

The Fashion-MNIST data set was designed to be a more challenging replacement for MNIST. The drop in ROC AUC is observed as the data set has more ambiguity between classes. One can notice the most significant performance drop for μ-SVM. While it is performing much better than the other methods for MNIST data set, it is visibly worse for Fashion-MNIST. β-CVAE ($\beta = 0.1$) yields the best detection results independent from chosen τ, compared to all baseline methods.

COIL-100 Benchmark Similarly to the previous experiment, the COIL-100 experiments used $\mathbb{D}_{\mathrm{KL}}$ as an anomaly score. The results, reported in Table 6.2, show that β-CVAE ($\beta = 7$) turns the $\mathbb{D}_{\mathrm{KL}}$ into a powerful anomaly indicator not only for cases of original style, as shown in the previous experiment, but also when samples are mislabelled. With an increasing number of base classes c, the problem becomes more challenging, but the ROC AUC for β-CVAE stays high. However, VAE is unable to indicate mislabelled samples regardless of the value of c. A simple label concatenation to the input vector will hardly influence the yield of $\mathbb{D}_{\mathrm{KL}}$ in vanilla VAE, suggesting that architectural changes are indeed necessary for successfully meeting the detection objective.

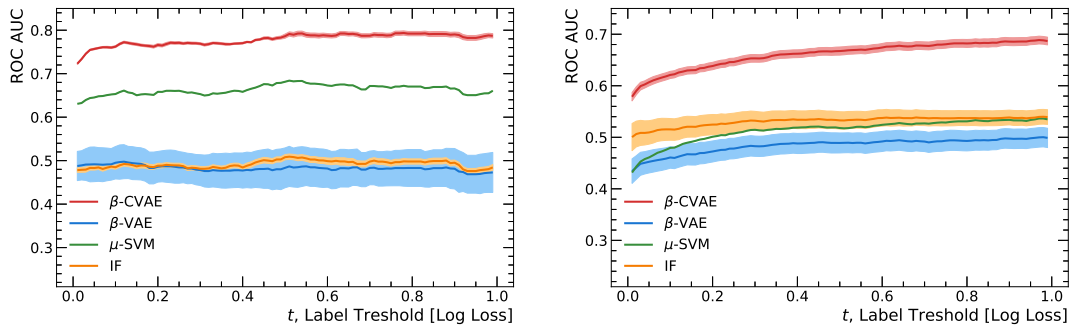


FIGURE 6.8: Reported ROC AUC for MNIST (left) Fashion-MNIST (right) data sets and different AD algorithms as a function of varying anomaly strictness τ based on LeNet-5 model classification log loss s. Overall classifier accuracy is 98.95% and 89.62% for MNIST and Fashion-MNIST respectively. The curves stay relatively flat due to the high performance of the classifier: most of the test samples have log loss smaller than 0.01. As a consequence, the test sets have low variability when values of τ change.

	$c = 10$	$c = 15$	$c = 20$	$c = 25$	$c = 30$	$c = 40$
β-CVAE	0.980	0.982	0.954	0.902	0.884	0.898
β-VAE	0.607	0.637	0.589	0.472	0.529	0.512

TABLE 6.2: ROC AUC for the COIL-100 data set based on the type B metric.

STRI and Trigger Rate Benchmarks The preliminary experiment examined sufficiency of the optimal reconstruction resolution from Section 6.3.1. The test was done using the synthetic data set. Because type A problems are generally easy to spot, as listed later in this section, the significance of β parameter is highlighted using type B detection only. Figure 6.9 shows the model's loss reported at the end of the training phase. With higher values of β, the $\mathbb{D}_{\mathrm{KL}}$ term is penalized more in the final reported loss. It also showes that the simplest introduction of the varying β parameter into equation 6.6 (β-CVAE), which weighs the importance of $\mathbb{D}_{\mathrm{KL}}$ term, yields unsurprisingly, varying type B detection results. Consequently, β is an important hyper-parameter that needs to be tuned.

Instead of using fix variance output the model should be allowed to learn it as it is beneficial for the target application. The trigger data is characterized by substantial noise, different for each trigger node. However the fix variance approach is tested for different values of $\beta \in [0.2, 8]$ with increments equal to 0.2, the results show that the AD-CVAE generally outperforms the β-CVAE. It appears that in our case, the need for tuning yet another hyper-parameter becomes obsolete with introduction of σ learning. For the values close to $\beta = 1.5$, the performance of β-CVAE marginally outperforms the ones of AD-CVAE.

Figure 6.10 shows the behaviour of the loss yield throughout the training. The curves for β-CVAE become flat after just a few epochs of training and have a steady decrease until final termination. On the contrary, AD-CVAE has visual steps at epochs 20 and 40 when model adjusts the trade-offs between ELBO components.

As discussed in Zhao et al. (2017) learning independent σ can unfortunately lead to inaccurate amortized inference distributions as the ELBO objective tends to sacrifice correct inference to overfit the training data, resulting in $\sigma = 0^{+}$. In such case the ELBO objective can be maximized even with a very inaccurate variational posterior $q_{\phi}(z|x)$. We suspect that in our

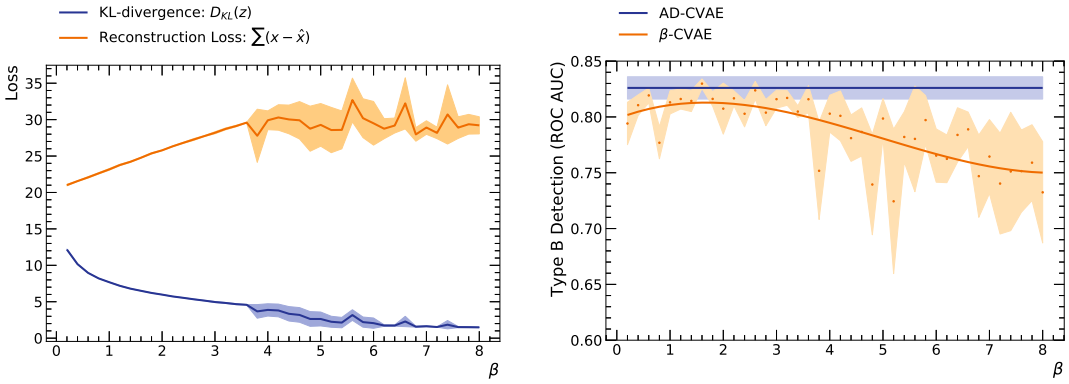


FIGURE 6.9: The final loss function values for a β-CVAE (left) and results of the type B detection problem for the synthetic data set as a function of varying β (right). The experiment was run five times for each of the chosen β values. The orange bands correspond to the standard deviation of the results. The dots correspond to the mean and the solid line to a polynomial fit of the means. The AD-CVAE ROC AUC stays constant as the training procedure does not depend on β hyper-parameter (see the loss in equation 6.4).

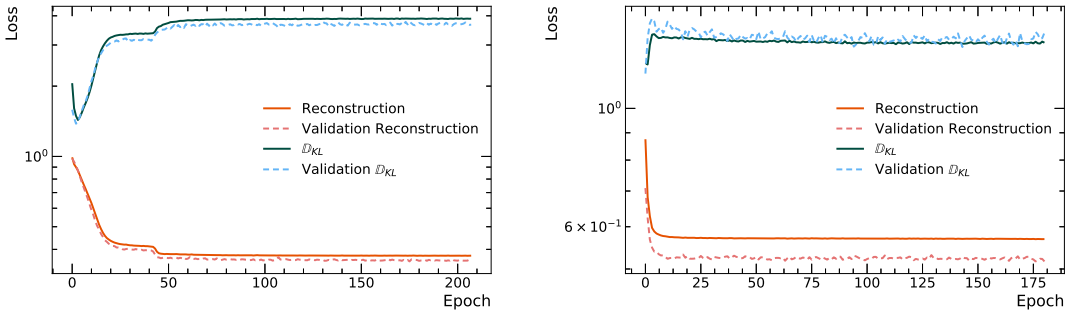


FIGURE 6.10: Loss yield as a function of the number of epochs in the training of the AD-CVAE model (left) and β-CVAE (right) used for the anomalous trigger rate detection. The two curves illustrate the behaviour of the training and validation data sets.

case this phenomena is not happening as the target application data is sufficiently noisy, preventing this unhealthy optimization. In order to prevent such undesirable behavior Lucas et al. (2019) suggested using β annealing Bowman et al. (2016), which helps to balance the objective and prevent $\sigma = 0^+$. An alternative is balancing the reconstruction and compression as proposed in Rezende and Viola (2018), through introduction of Lagrange multipliers which are optimized following a moving average of the constraint vector.

The full results for synthetic and trigger rates data sets are listed in Tables 6.3 and 6.4 for type A and type B problems, respectively. AD-CVAE is compared with β-CVAE (using fixed output variance $\sigma = 1$) with two proposed values of β and the vanilla VAE. $\beta = 1.5$ was suggested by the test results seen in Figure 6.9; $\beta = 4$ was suggested by Higgins et al. (2017). The different levels of s test different severity of problems. For visual comparison and chosen values of s, the ROC curves corresponding to both the synthetic and trigger rate AD problems are shown in Figures 6.11 and 6.12. Legacy requirements of the monitoring software motivate the values of s.

Given the high order of the deviation on type A anomalies, both the CVAE and VAE easily spot those types of problems. In the context of the hierarchical structures and the target

Synthetic

	AD-CVAE	β-CVAE ($\beta = 1.5$)	β-CVAE ($\beta = 4$)	VAE
$s = 1\sigma$	0.903 ± 0.010	0.792 ± 0.001	0.790 ± 0.002	0.823 ± 0.006
$s = 2\sigma$	0.959 ± 0.006	0.965 ± 0.000	0.962 ± 0.001	0.985 ± 0.004
$s = 3\sigma$	0.983 ± 0.004	0.996 ± 0.000	0.996 ± 0.000	0.998 ± 0.001
$s = 4\sigma$	0.992 ± 0.002	0.999 ± 0.000	0.999 ± 0.000	0.999 ± 0.000
$s = 5\sigma$	0.996 ± 0.002	0.999 ± 0.000	0.999 ± 0.000	0.999 ± 0.000

Trigger rates

	AD-CVAE	β-CVAE ($\beta = 1.5$)	β-CVAE ($\beta = 4$)	VAE
$s = 1\sigma$	0.592 ± 0.017	0.564 ± 0.002	0.543 ± 0.001	0.541 ± 0.002
$s = 2\sigma$	0.798 ± 0.018	0.778 ± 0.002	0.765 ± 0.002	0.758 ± 0.001
$s = 3\sigma$	0.915 ± 0.016	0.918 ± 0.002	0.915 ± 0.001	0.910 ± 0.001
$s = 4\sigma$	0.965 ± 0.011	0.971 ± 0.001	0.973 ± 0.000	0.971 ± 0.000
$s = 5\sigma$	0.984 ± 0.009	0.991 ± 0.001	0.992 ± 0.001	0.992 ± 0.000

TABLE 6.3: ROC AUC for the type A detection for different severity of the problem (the higher the s, the higher severity), see table 6.1. The metric used is the average infinity norm of the reconstruction loss $||\frac{1}{\sigma}(x - \hat{x})^2||_\infty$.

Synthetic

	AD-CVAE	β-CVAE ($\beta = 1.5$)	β-CVAE ($\beta = 4$)	VAE
$s = 1\sigma$	0.614 ± 0.005	0.614 ± 0.005	0.604 ± 0.024	0.553 ± 0.011
$s = 2\sigma$	0.743 ± 0.005	0.746 ± 0.014	0.731 ± 0.049	0.631 ± 0.024
$s = 3\sigma$	0.827 ± 0.011	0.823 ± 0.022	0.809 ± 0.059	0.683 ± 0.027
$s = 4\sigma$	0.871 ± 0.021	0.872 ± 0.027	0.859 ± 0.057	0.723 ± 0.022
$s = 5\sigma$	0.893 ± 0.027	0.902 ± 0.030	0.887 ± 0.056	0.743 ± 0.022

Trigger rates

	AD-CVAE	β-CVAE ($\beta = 1.5$)	β-CVAE ($\beta = 4$)	VAE
$s = 1\sigma$	0.661 ± 0.002	0.700 ± 0.003	0.623 ± 0.003	0.626 ± 0.002
$s = 2\sigma$	0.816 ± 0.007	0.820 ± 0.005	0.673 ± 0.002	0.665 ± 0.007
$s = 3\sigma$	0.888 ± 0.009	0.901 ± 0.005	0.687 ± 0.006	0.675 ± 0.008
$s = 4\sigma$	0.918 ± 0.012	0.930 ± 0.007	0.683 ± 0.013	0.670 ± 0.011
$s = 5\sigma$	0.929 ± 0.014	0.937 ± 0.008	0.686 ± 0.018	0.659 ± 0.018

TABLE 6.4: ROC AUC for the type B detection for different severity of the problem (the higher the s, the higher severity), see table 6.1. The metric used is the mean μ of KL divergence $\mathbb{D}_{\mathrm{KL}}$.

application, an algorithm needs to model mapping from a single input to multiple possible outputs. In this context, as argued in Sohn et al. (2015) model needs to make different predictions. However, a generalization beyond nominal behaviour (generalizing to rare or anomalous samples) may yield high FN rate as CVAE may learn to reconstruct anomalous samples accurately. The results show that this over-generalization is prevented.

The type B detection provides good results outperforming vanilla VAE baseline and confirming that CVAEs are suitable for distinguishing between anomalous and noisy behaviour. As argued before, the marginally superior results of β-CVAE ($\beta = 1.5$) over AD-CVAE are likely the result of the complexity of the learning gradient. However, those marginal gains come at the expense of much longer training times that need to account in the non-trivial hyperparameter scan. For instance, it can be noticed that for $\beta = 4$, the results for synthetic data set are competitive, while for the trigger rate data set, the performance drops significantly.

The performance of the algorithm on CMS data set is matching the performance reported

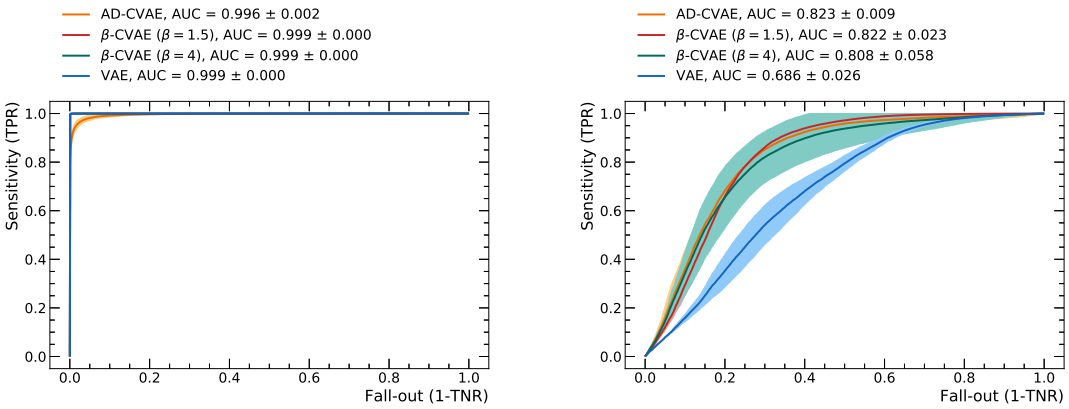


FIGURE 6.11: The ROC curves for two AD problems using synthetic test data set for type A (left) and type B detection (right). The bands correspond to variance computed after running the experiment five times using random weight initialization. Anomaly score for type B is computed for $s = 3\sigma$ using mean $\mathbb{D}_{\mathrm{KL}}$ of z. Anomaly score for type A problem is computed for $s = 5\sigma$ using decoder outputs: μ and σ of each feature.

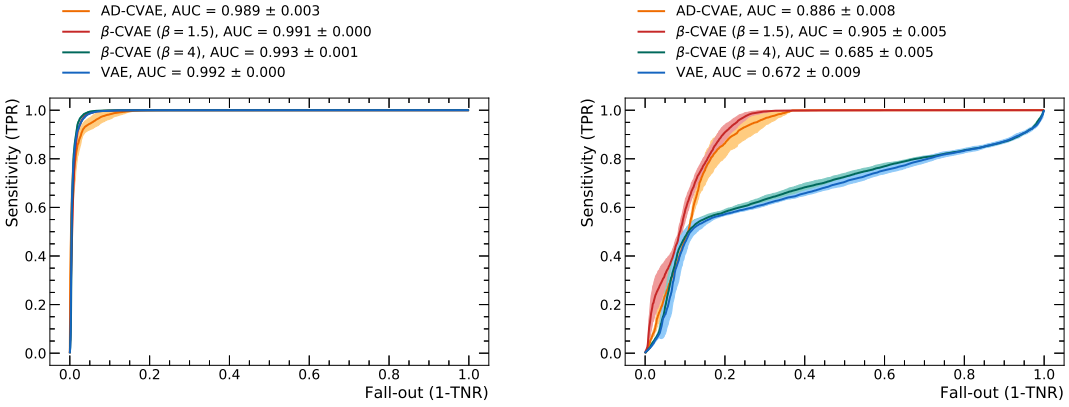


FIGURE 6.12: The ROC curves for two AD problems using trigger rate data set for type A (left) and type B detection (right). The bands correspond to variance computed after running the experiment five times using random weight initialization. Anomaly score for type B is computed for $s = 3\sigma$ using mean $\mathbb{D}_{\mathrm{KL}}$ of z. Anomaly score for type A problem is computed for $s = 5\sigma$ using decoder outputs: μ and σ of each feature.

for the synthetic one. Hence, the AD-CVAE can learn representations from input data in a non-linear way. As described at the beginning of this chapter, the CMS experiment does not provide any tools to track problems falling into type B category at this time. Thus it is not possible to compare the obtained results to a production baseline. Inference time is negligible in the context of the target application. The upper limit of $\sim$ 23 s is easily met by the algorithm even on the commercial CPU with prediction time in order of milliseconds.

Figure 6.13 shows how changing the size of vectors k and u influences both types of detection. For type A, all models, regardless of values of m (size of vector k) and o (size of vector u) report comparable results. Instead, the sensitivity for type B drops both for increasing values of m and $\frac{m}{o}$ ratio. That suggests two critical implementation assumptions. First, the o should approximately equal to m. Second, the size of the configuration group, relative to sample size ($\frac{n}{m}$) should be kept high.

6.6 Conclusions and Practical Considerations

This chapter showed how anomalous samples could be identified using CVAE-based architecture. The specific case of CMS TRM has been considered to extend current monitoring functionality. As suggested in the chapter's introduction, an algorithm was expected to have excellent detection performance, be robust and have a straightforward implementation. AD-CVAE does qualify for the final solution as it provides superior detection performance. It is based on the variational framework, which generalizes well beyond training samples. Finally, the promise of the use of only one algorithm across the full configuration guarantees maintenance, deployment and inference simplicity. Subsequent studies foresee using a full configuration of the CMS trigger system.

The method was demonstrated not to be bound to CMS experiment specifics and has the potential to work across different domains. Furthermore, a hyper-parameter scan was only performed using random search for all of the experiments. Thus the results are expected to get better if further optimized. An appealing extension of the method would be to take disentanglement a step further and learn the correct encoding of unknown factors of variations in the latent space.

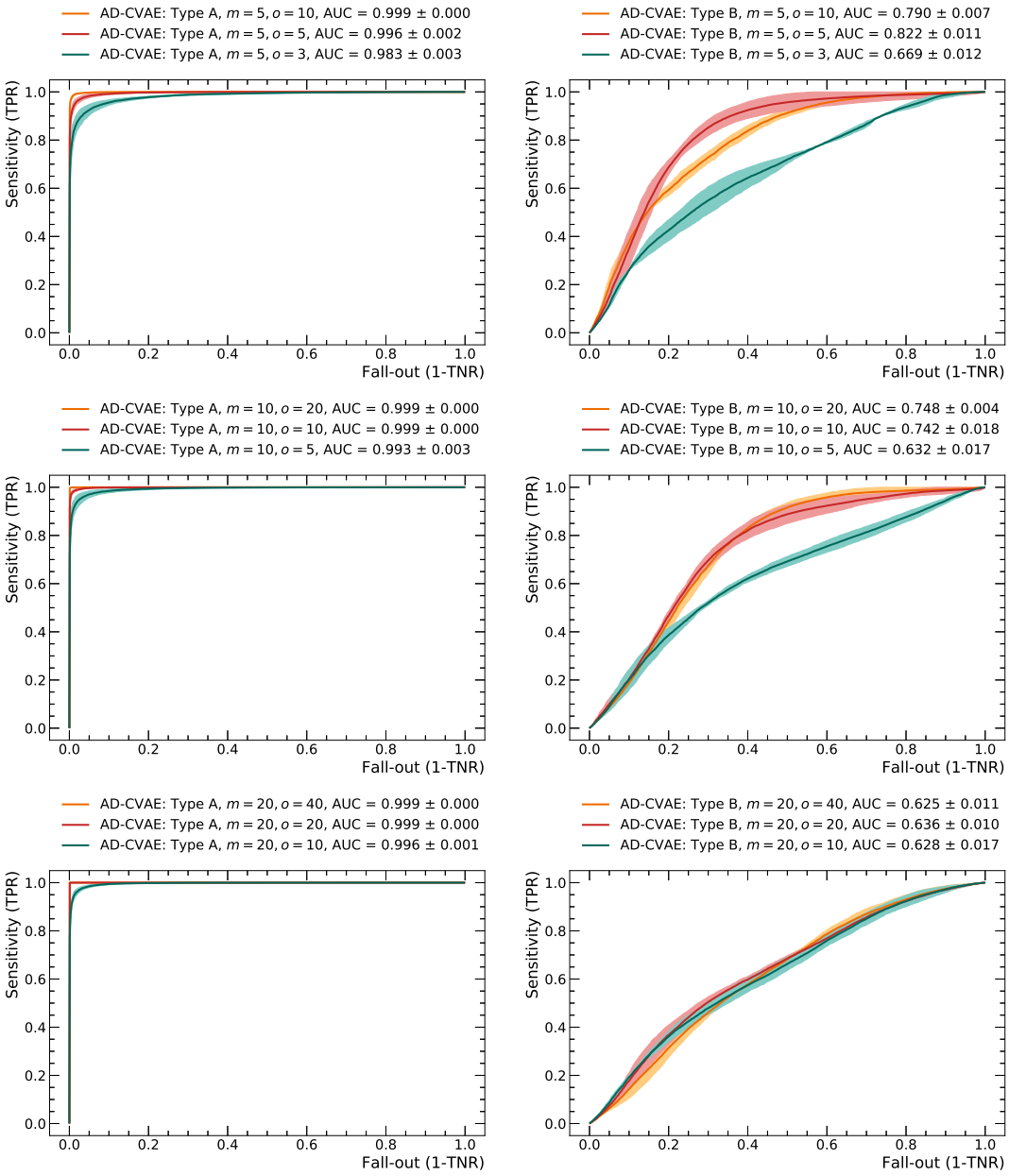


FIGURE 6.13: The ROC curves for two AD problems using synthetic test data set for type A (left) and type B detection (right) run for different size of k and u vectors. The bands correspond to variance computed after running the experiment five times using random weight initialization. Anomaly score for type B is computed for $s = 3\sigma$ using mean $\mathbb{D}_{\mathrm{KL}}$ of z. Anomaly score for type A problem is computed for $s = 5\sigma$ using decoder outputs: μ and σ of each feature.

CHAPTER 7

Conclusions

The present dissertation discusses the application of novel techniques for the automation of AD procedures in HEP experiments. The CMS DQM use case is illustrated as a paradigm of monitoring infrastructure in HEP collaborations, challenging its established protocols for quality control via the application of DNNs.

This work proposes novel approaches to the different DQM application realms, taking into account the corresponding constraints and requirements: online monitoring, offline certification of the data as usable for physics analysis, and, finally, the live monitoring of data acquisition rates.

For the online monitoring of the sub-detector components, required to pinpoint problematic detector components with low latency, a classifier capable of detecting the known anomalous behaviours is proposed, together with methods for expanding current monitoring coverage detecting novel failure modes. The results show unprecedented efficiency of CNNs on currently tracked failure modes. The work covered aspects related to model retraining strategies and interpretation of the results, which are of paramount importance in a system which will need to be operated for years by field experts with limited ML expertise. The applicability of autoencoders was demonstrated covering the offline monitoring for identification of novel and emerging problems behaviours on a large number of observables with fine time granularity and potentially low statistics. Finally, in the application of AD on hierarchically structured data for the monitoring of acquisition rates a new method was introduced, AD-CVAE, which was shown to be a suitable solution for detecting anomalies affecting features in the same configuration groups.

The methods proposed here are general enough to be applicable beyond the physics domain. The results demonstrate once more that the DNN AD provides a breakthrough for complex and high dimensional problems in infrastructure monitoring.

The experimental results have been very well received by the domain experts. Some of the proposed models have already been integrated and deployed in the production CMS online DQM infrastructure. A generalization of the strategies proposed throughout this thesis paves the way to full automation of the quality assessment for HEP experiments. The relevance of this seminal work is confirmed by the fact that the CMS collaboration has indeed

instantiated a structured effort to continue exploiting DL methods in DQM context: a group of researchers (including the author) is actively pursuing this task.

Finally, this exploratory work and its success story have contributed in promoting ML AD in the highly sceptical HEP community for application beyond the DQM itself: the successes in quality control sparkled interest in the application of similar techniques to other challenges, e.g. searches of physics beyond the SM with ML AD.

Acronyms

μ-SVM One-Class Support Vector Machine.

k-NN K Nearest Neighbors.

AD Anomaly Detection.

AI Artificial Intelligence.

ALICE A Large Ion Collider Experiment.

ANN Artificial Neural Networks.

AOD Analysis Object Data.

ASIC Application Specific Integrated Circuit.

ATLAS A Toroidal LHC ApparatuS.

AUC Area Under the Curve.

AVB Adversarial Variational Bayes.

BBVI Black Box Variational Inference.

CAVI Coordinate Ascent Variation Inference.

CERN European Organization for Nuclear Research.

CMS Compact Muon Solenoid.

CNN Convolutional Neural Network.

COIL Columbia Object Image Library.

CP Charge Conjugation Parity.

CVAE Conditional Variational Autoencoder.

DC Data Certification.

DGM Deep Latent Gaussian Model.

DL Deep Learning.

DNN Deep Neural Network.

DQM Data Quality Monitoring.

DT Drift Tube.

EB Exabyte.

ECAL Electromagnetic Calorimeter.

ELBO Evidence Lower Bound.

EM Expectation Maximisation.

FN False Negatives.

FNR False Negative Rate.

FP False Positive.

FPGA Field Programmable Gate Array.

FPR False Positive Rate.

GAN Generative Adversarial Networks.

GEM Gas Electron Multiplier.

GMM Gaussian Mixture Model.

HCAL Hadron Calorimeter.

HEP High Energy Physics.

HLT High Level Trigger.

ICA Independent Component Analysis.

IF Isolation Forest.

IWAE Importance Weighted Variational Autoencoder.

KL Kullback-Leibler.

L1 Trigger Level 1 Trigger.

LHC Large Hadron Collider.

LHCb Large Hadron Collider beauty.

LOCI Local Correlation Integral.

LOF Local Outlier Factor.

LS Lumisection.

MC Monte Carlo.

MCMC Markov Chain Monte Carlo.

ML Machine Learning.

MLE Maximum Likelihood Estimation.

MNIST Modified National Institute of Standards and Technology.

MSE Mean Squared Error.

OCC One-Class Classification.

OOD Out of Distribution.

PCA Principal Components Analysis.

PCC Pearson Correlation Coefficient.

PD Primary Dataset.

PDF Probability Density Function.

PPV Positive Predictive Value.

PR Precision-Recall.

PS Proton Synchrotron.

PSB PS Booster.

PU pile-up.

RBF Radial Basis Function.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic.

SGD Stochastic Gradient Descent.

SM Standard Model.

SNN Shallow Neural Network.

SPS Supper Proton Synchrotron.

SSND Semi-Supervised Novelty Detection.

STRI Synthetic Trigger.

SVDD Support Vector Data Description.

SVI Stochastic Variational Inference.

SVM Support Vector Machine.

TC Total Correlation.

TN True Negative.

TNR True Negative Rate.

TP True Positive.

TPR True Positive Rate.

TRM Trigger Rate Monitoring.

VAE Variational Autoencoder.

VB Variational Bayes.

VI Variational Inference.

Bibliography

Aamodt, K., Quintana, A. A., Achenbach, R., Acounis, S., Adamová, D., Adler, C., Aggarwal, M., Agnese, F., Rinella, G. A., Ahammed, Z., et al. (2008). The ALICE experiment at the CERN LHC. *Journal of Instrumentation*, 3(08):S08002–S08002.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, abs/1603.04467.

Abe, N., Zadrozny, B., and Langford, J. (2006). Outlier Detection by Active Learning. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 504–509, New York, NY, USA. Association for Computing Machinery.

Adam-Bourdarios, C., Cowan, G., Germain, C., Guyon, I., Kégl, B., and Rousseau, D. (2014). The Higgs Boson Machine Learning Challenge. In *Proceedings of the 2014 International Conference on High-Energy Physics and Machine Learning - Volume 42*, HEPML'14, page 19–55. JMLR.org.

Aggarwal, C. C. (2014). *Data Classification: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition.

Aggarwal, C. C. (2016). *Outlier Analysis*. Springer Publishing Company, Incorporated, 2nd edition.

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, page 420–434, Berlin, Heidelberg. Springer-Verlag.

Aizenberg, I. N., Aizenberg, N. N., and Vandewalle, J. P. (2000). *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Kluwer Academic Publishers, USA.

Alain, G. and Bengio, Y. (2014). What Regularized Auto-Encoders Learn from the Data-Generating Distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593.

An, J. and Cho, S. (2015). Variational Autoencoder based Anomaly Detection using Reconstruction Probability. Technical report, SNU Data Mining Center.

Andrews, J., Morton, E., and Griffin, L. (2016). Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing*, 6:21.

Anscombe, F. J. (1960). Rejection of Outliers. *Technometrics*, 2(2):123–146.

Atha, D. J. and Jahanshahi, M. R. (2018). Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5):1110–1128.

Atlas, L. E., Cohn, D. A., and Ladner, R. E. (1990). Training Connectionist Networks with Queries and Selective Sampling. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 566–573. Morgan-Kaufmann.

ATLAS Collaboration (2008). The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003.

Azzolini, V., Borisyak, M., Cerminara, G., Derkach, D., Franzoni, G., Guio, F. D., Koval, O., Pierini, M., Pol, A., Ratnikov, F., Siroky, F., Ustyuzhanin, A., and Vlimant, J.-R. (2018). Deep learning for inferring cause of data anomalies. *Journal of Physics: Conference Series*, 1085:042015.

Barnett, V. and Lewis, T. (1978). *Outliers in statistical data.* John Wiley & Sons Ltd., 2nd edition edition.

Bay, S. D. and Schwabacher, M. (2003). Mining Distance-Based Outliers in near Linear Time with Randomization and a Simple Pruning Rule. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 29–38, New York, NY, USA. Association for Computing Machinery.

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

Bengio, Y. and LeCun, Y. (2007). Scaling Learning Algorithms Towards AI. In Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., editors, *Large Scale Kernel Machines*. MIT Press, Cambridge, MA.

Bernreuther, W. (2002). *CP Violation and Baryogenesis*, pages 237–293. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bertone, G., Hooper, D., and Silk, J. (2004). Particle Dark Matter: Evidence, Candidates and Constraints. *Physics Reports*, 405.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Blanchard, G., Lee, G., and Scott, C. (2010). Semi-supervised novelty detection. *J. Mach. Learn. Res.*, 11:2973–3009.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U., and Zieba, K. (2016). VisualBackProp: efficient visualization of CNNs. *arXiv preprint arXiv:1611.05418*.

Borisyak, M., Ratnikov, F., Derkach, D., and Ustyuzhanin, A. (2017). Towards automation of data quality system for CERN CMS experiment. *IOP Conf. Ser J Phys Confer Ser*, 898(9):092041.

Borowiec, S. (2016). Alphago seals 4-1 victory over go grandmaster lee sedol. *The Guardian*, 15.

Bottou, L. and LeCun, Y. (2003). Large scale online learning. In *Advances in Neural Information Processing Systems 16 NIPS*, pages 217–224.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: Identifying Density-based Local Outliers. *SIGMOD Rec.*, 29(2):93–104.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. (2016). Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-vae. *CoRR*, abs/1804.03599.

Butepage, J., He, J., Zhang, C., Sigal, L., and Mandt, S. (2018). Informed priors for deep representation learning. In *Symposium on Advances in Approximate Bayesian Inference*.

CERN (2016). CERN's Accelerator Complex.

Chalapathy, R. and Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

Chalapathy, R., Menon, A. K., and Chawla, S. (2018). Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Chatrchyan, S. et al. (2008). The CMS experiment at the CERN LHC. *JINST*, 3:S08004.

Chatrchyan, S. et al. (2012). Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716(1):30–61.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 2610–2620. Curran Associates, Inc.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. (2017). Variational lossy autoencoder. In *International Conference on Learning Representations, ICLR 2017*.

Cherief-Abdellatif, B.-E. (2019). Consistency of ELBO maximization for model selection. In Ruiz, F., Zhang, C., Liang, D., and Bui, T., editors, *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*, volume 96 of *Proceedings of Machine Learning Research*, pages 11–31. PMLR.

Choi, H., Jang, E., and Alemi, A. A. (2018). WAIC, but Why? Generative Ensembles for Robust Anomaly Detection. *arXiv preprint arXiv:1810.01392*.

Chollet, F. et al. (2015). Keras.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2980–2988. Curran Associates, Inc.

CMS Collaboration (1997a). The CMS electromagnetic calorimeter project: Technical Design Report. *Technical Design Report CMS. Geneva: CERN*, pages 97–33.

CMS Collaboration (1997b). The CMS hadron calorimeter project: Technical Design Report. *Technical Design Report CMS. CERN, Geneva*.

CMS Collaboration (1997c). *The CMS magnet project: Technical Design Report*. Technical Design Report CMS. CERN, Geneva.

CMS Collaboration (2000a). Addendum to the CMS Tracker TDR. *CERN/LHCc*, 16:2000.

CMS Collaboration (2000b). The TriDAS project: Technical Design Report, Vol. 1: The Trigger Systems. Technical report, CERN-LHCC-2000-038.

CMS Collaboration (2002). The TriDAS project: Technical Design Report, Vol. 2: Data Acquisition and High-Level Trigger. *CERN/LHCC*, 26(6.1).

CMS Collaboration (2010). Calibration of the CMS drift tube chambers and measurement of the drift velocity with cosmic rays. *Journal of Instrumentation*, 5(03):T03016.

Council, C. (2013). The European Strategy for Particle Physics Update 2013.

Cowan, G., Cranmer, K., Gross, E., and Vitells, O. (2011). Asymptotic formulae for likelihood-based tests of new physics. *The European Physical Journal C*, 71(2):1554.

Craig, N. (2013). The State of Supersymmetry after Run I of the LHC. *arXiv preprint arXiv:1309.0528*.

Cremer, C., Li, X., and Duvenaud, D. (2018). Inference Suboptimality in Variational Autoencoders. In *International Conference on Machine Learning*.

Cremer, C., Morris, Q., and Duvenaud, D. (2017). Reinterpreting Importance-Weighted Autoencoders. In *Workshop at the International Conference on Learning Representations*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Davis, S. R. (2016). Interactive Slice of the CMS detector.

De Guio, F. (2014). The CMS data quality monitoring software: experience and future prospects. *Journal of Physics: Conference Series*, 513(3):032024.

Dellaert, F. (2002). The expectation maximization algorithm. Technical report, Georgia Institute of Technology.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd.

Englert, F. and Brout, R. (1964). Broken symmetry and the mass of gauge vector mesons. *Physical Review Letters*, 13(9):321.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 255–262, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Eslami, S. A., Rezende, D. J., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., et al. (2018). Neural scene representation and rendering. *Science*, 360(6394):1204–1210.

Fabius, O., van Amersfoort, J. R., and Kingma, D. P. (2015). Variational recurrent auto-encoders. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Fu, M. C. (2006). Chapter 19 gradient estimation. In Henderson, S. G. and Nelson, B. L., editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, pages 575 – 616. Elsevier.

Gammerman, A., Vovk, V., and Vapnik, V. (1998). Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 148–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Gemici, M., Hung, C.-C., Santoro, A., Wayne, G., Mohamed, S., Rezende, D. J., Amos, D., and Lillicrap, T. (2017). Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*.

Ghahramani, Z. and Beal, M. J. (2001). Propagation algorithms for variational bayesian learning. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 507–513. MIT Press.

Gibson, W. (1960). Nonlinear factors in two dimensions. *Psychometrika*, 25:381–392.

Gill, K. and EP-CMX (2017). CMS pixel upgrade: a truly global endeavor.

Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173.

Gonzalez-Garcia, M., Maltoni, M., and Schwetz, T. (2016). Global analyses of neutrino oscillation experiments. *Nuclear Physics B*, 908:199–217.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning (pages 499-523)*. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. (2015). DRAW: A Recurrent Neural Network For Image Generation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1462–1471, Lille, France. PMLR.

Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. (2014). Deep autoregressive networks. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1242–1250, Bejing, China. PMLR.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.

Gulrajani, I., Kumar, K., Ahmed, F., Taiga, A. A., Visin, F., Vazquez, D., and Courville, A. (2017). PixelVAE: A Latent Variable Model for Natural Images. In *5th International Conference on Learning Representations*.

Hanser, D. A. (2006). *Architecture of France*. Greenwood Publishing Group.

Hawkins, D. M. (1980). *Identification of outliers*, volume 11. Springer.

Hawkins, S., He, H., Williams, G., and Baxter, R. (2002). Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings to ICCV*, pages 1026–1034.

Hendrycks, D., Mazeika, M., and Dietterich, T. G. (2018). Deep anomaly detection with outlier exposure. *CoRR*, abs/1812.04606. ICLR19.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Higgs, P. W. (1964). Broken symmetries and the masses of gauge bosons. *Physical Review Letters*, 13(16):508.

Hinneburg, A., Aggarwal, C. C., and Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In *26th Internat. Conference on Very Large Databases*, pages 506–515.

Hinton, G. E. (1990). Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier.

Ho-Kim, Q. and Pham, X.-Y. (2013). *Elementary particles and their interactions: concepts and phenomena*. Springer Science & Business Media.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347.

Hoffman, M. D. and Johnson, M. J. (2016). Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1.

Hou, X., Shen, L., Sun, K., and Qiu, G. (2017). Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141. IEEE.

Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.

Hyvärinen, A., Sasaki, H., and Turner, R. E. (2019). Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 859–868.

Jain, A. K. and Dubes, R. C. (1988). Algorithms for clustering data. *Englewood Cliffs: Prentice Hall, 1988*.

Japkowicz, N., Myers, C., Gluck, M., et al. (1995). A novelty detection approach to classification. In *IJCAI*, volume 1, pages 518–523.

Johnson, M. J., Duvenaud, D. K., Wiltschko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations and fast inference. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2946–2954. Curran Associates, Inc.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kang, D. and Lee, J. (2018). GEM DQM with Machine Learning.

Karimäki, V., Mannelli, M., Siegrist, P., Breuker, H., Caner, A., Castaldi, R., Freudenreich, K., Hall, G., Horisberger, R., Huhtinen, M., and Cattai, A. (1997). *The CMS tracker system project: Technical Design Report*. Technical Design Report CMS. CERN, Geneva.

Kawachi, Y., Koizumi, Y., and Harada, N. (2018). Complementary set variational autoencoder for supervised anomaly detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2366—-2370.

Kearns, M. J. (1990). *The computational complexity of machine learning*. MIT press.

Khachatryan, V. et al. (2015). Precise determination of the mass of the Higgs boson and tests of compatibility of its couplings with the standard model predictions using proton collisions at 7 and 8 TeV. *The European Physical Journal C*, 75(5):212.

Khachatryan, V. et al. (2017). The CMS trigger system. *JINST*, 12(01):P01020.

Khan, S. S. and Madden, M. G. (2014). One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374.

Khemakhem, I., Kingma, D. P., and Hyvärinen, A. (2019). Variational autoencoders and nonlinear ica: A unifying framework. *arXiv preprint arXiv:1907.04809*.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658, Stockholmsmässan, Stockholm Sweden. PMLR.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4743–4751, USA. Curran Associates Inc.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kliger, M. and Fleishman, S. (2018). Novelty detection with gan. *arXiv preprint arXiv:1802.10560*.

Krishnan, R., Liang, D., and Hoffman, M. (2018). On the challenges of learning with inference networks on sparse, high-dimensional data. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 143–151, Playa Blanca, Lanzarote, Canary Islands. PMLR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Kwon, D., Natarajan, K., Suh, S. C., Kim, H., and Kim, J. (2018). An empirical study on network anomaly detection using convolutional neural networks. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1595–1598.

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S., and Kavsek, B. (2000). Informal identification of outliers in medical data. In *Fifth international workshop on intelligent data analysis in medicine and pharmacology*, volume 1, pages 20–24.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc.

LHCb Collaboration (2008). The LHCb Detector at the LHC. *JINST*, 3(LHCb-DP-2008-001. CERN-LHCb-DP-2008-001):S08005. Also published by CERN Geneva in 2010.

Li, H. (2010). Research and implementation of an anomaly detection model based on clustering analysis. In *2010 International Symposium on Intelligence Information Processing and Trusted Computing*, pages 458–462.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124, Long Beach, California, USA. PMLR.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., and Lloret, J. (2017). Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors*, 17(9):1967.

Louizos, C. and Welling, M. (2016). Structured and efficient variational deep learning with matrix gaussian posteriors. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ICML'16, pages 1708–1716. JMLR.org.

Lucas, J., Tucker, G., Grosse, R., and Norouzi, M. (2019). Understanding posterior collapse in generative latent variable models. In *7th International Conference on Learning Representations, ICLR 2019*.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Magnan, A.-M. (2017). Hgcal: a high-granularity calorimeter for the endcaps of cms at hl-lhc. *Journal of Instrumentation*, 12(01):C01042.

Mansbridge, A., Fierimonte, R., Feige, I., and Barber, D. (2019). Improving latent variable descriptiveness by modelling rather than ad-hoc factors. *Machine Learning*, 108(8):1601–1611.

Marcastel, F. C. G. D. s. (2013). CMS Tent Point 5 Cessy.

Mathieu, M. F., Zhao, J. J., Zhao, J., Ramesh, A., Sprechmann, P., and LeCun, Y. (2016). Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mescheder, L., Nowozin, S., and Geiger, A. (2017). Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2391–2400, International Convention Centre, Sydney, Australia. PMLR.

Minsky, M. L. (1967). *Computation: Finite and infinite machines*. Prentice-Hall Englewood Cliffs.

MissMJ (2006). Standard model of elementary particles.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, ICML'10, pages 807–814, USA. Omnipress.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2018). Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. (2019). Detecting out-of-distribution inputs to deep generative models using a test for typicality. *arXiv preprint arXiv:1906.02994*.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia Object Image Library (COIL-100).

Paul Scherrer Institute (2017). Silicon pixel barrel detector successfully installed in the CMS experiment.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572.

Ramaswamy, S., Rastogi, R., and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29(2):427–438.

Ramotsoela, D., Abu-Mahfouz, A., and Hancke, G. (2018). A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study. *Sensors*, 18(8):2491.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black Box Variational Inference. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. PMLR.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2006). Efficient learning of sparse representations with an energy-based model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 1137–1144.

Rapsevicius, V. et al. (2011). CMS Run Registry: Data certification bookkeeping and publication system. *Journal of Physics: Conference Series*, 331(4):042038.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Rezende, D. J. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning - Volume 32*, ICML, pages 1278–1286.

Rezende, D. J. and Viola, F. (2018). Taming vaes. *arXiv preprint arXiv:1810.00597*.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 833–840.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Roberts, A., Engel, J., and Eck, D. (2017). Hierarchical variational autoencoders for music. In *NIPS Workshop on Machine Learning for Creativity and Design*.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

Rovelli, C. (2011). Zakopane lectures on loop gravity. *arXiv preprint arXiv:1102.3660*.

Rovere, M. (2015). The Data Quality Monitoring Software for the CMS experiment at the LHC. *Journal of Physics: Conference Series*, 664(7):072039.

Roy, N. and McCallum, A. (2001). Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448.

Rubenstein, P. K., Schölkopf, B., and Tolstikhin, I. O. (2018). Learning disentangled representations with wasserstein auto-encoders. In *6th International Conference on Learning Representations, ICLR 2018, Workshop Track Proceedings*.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617.

Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schneider, M. (2018). The Data Quality Monitoring software for the CMS experiment at the LHC: past, present and future. In *Proceedings to CHEP 2018*.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. (2016). Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR, abs/1610.02391*, 7.

Settles, B. (2009). Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Settles, B., Craven, M., and Ray, S. (2008). Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM.

Shafaei, A., Schmidt, M., and James, J. (2018). Little. does your model know the digit 6 is not a cat? a less biased evaluation of outlier detectors. *arXiv preprint arXiv:1809.04729*.

Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810.

Siddharth, N., Paige, B., van de Meent, J.-W., Desmaison, A., Goodman, N. D., Kohli, P., Wood, F., and Torr, P. H. (2017). Learning disentangled representations with semi-supervised deep generative models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5927–5937, USA. Curran Associates Inc.

Simard, P. Y., LeCun, Y. A., Denker, J. S., and Victorri, B. (1998). Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Sirunyan, A. M. et al. (2018). Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV. *arXiv preprint arXiv:1804.04528*.

Snoek, J., Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J., Ren, J., and Nado, Z. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980.

Sobel, I. (1990). An isotropic 3×3 image gradient operator. *Machine vision for three-dimensional scenes*, pages 376–379.

Soha, A. (2011). Web Based Monitoring in the CMS Experiment at CERN. Technical Report CMS-CR-2011-013, CERN, Geneva.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491.

Sønderby, C., Raiko, T., Maaløe, L., Sønderby, S., and Winther, O. (2016). How to train deep variational autoencoders and probabilistic ladder networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*.

Song, X., Wu, M., Jermaine, C., and Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5).

Stankevicius, M., Marcinkevicius, V., and Rapsevicius, V. (2018). Comparison of supervised machine learning techniques for cern cms offline data certification. In *Doctoral Consortium/Forum@ DB&IS*, pages 170–176.

Suh, S., Chae, D. H., Kang, H.-G., and Choi, S. (2016). Echo-state conditional variational autoencoder for anomaly detection. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1015–1022. IEEE.

Sun, J. (2019). Feedforward neural networks.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Sölch, M., Bayer, J., Ludersdorfer, M., and van der Smagt, P. (2016). Variational Inference for On-line Anomaly Detection in High-Dimensional Time Series. *arxiv:1602.07109*.

Tax, D. M. and Duin, R. P. (1999a). Data domain description using support vectors. In *ESANN*, volume 99, pages 251–256.

Tax, D. M. and Duin, R. P. (1999b). Support vector domain description. *Pattern recognition letters*, 20(11-13):1191–1199.

Tax, D. M. J. (2001). *One-class classification: Concept learning in the absence of counter-examples*. PhD thesis, Technische Universiteit Delft.

The LHC Study Group (1995). The Large Hadron Collider, Conceptual Design. Technical report, CERN/AC/95-05 (LHC) Geneva.

Theiler, J. P. and Cai, D. M. (2003). Resampling approach for anomaly detection in multispectral images. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery IX*, volume 5093, pages 230–240. International Society for Optics and Photonics.

Tishby, N. and Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop*, pages 1–5.

Titsias, M. K. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1971–II–1980. JMLR.org.

Tomanek, K. and Olsson, F. (2009). A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 45–48. Association for Computational Linguistics.

Tricomi, A. (2014). Upgrade of the cms tracker. *Journal of Instrumentation*, 9(03):C03041.

Tschannen, M., Bachem, O., and Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *NeurIPS*.

Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. In Barber, D., Cemgil, T., and Chiappa, S., editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press.

Tuura, L., Eulisse, G., and Meyer, A. (2010). CMS data quality monitoring web service. In *Journal of Physics: Conference Series*, volume 219, page 072055. IOP Publishing.

van Veen, F. and Leijnen, S. (2019). The neural network zoo.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.

Wang, X., Du, Y., Lin, S., Cui, P., and Yang, Y. (2019). Self-adversarial variational autoencoder with gaussian anomaly prior distribution for anomaly detection. *CoRR*, abs/1903.00904.

Wang, Y. and Blei, D. M. (2019). Frequentist consistency of variational bayes. *Journal of the American Statistical Association*, 114(527):1147–1161.

Watanabe, S. (1960). Information theoretical analysis of multivariate correlation. *IBM J. Res. Dev.*, 4(1):66–82.

White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5):1097–1126.

Widrow, B. and Hoff, M. E. (1962). Associative storage and retrieval of digital information in networks of adaptive "neurons". In *Biological Prototypes and Synthetic Systems*, pages 160–160. Springer.

Wightman, A. et al. (2018). Tools for Trigger Rate Monitoring at CMS.

Wu, M. and Jermaine, C. (2006). Outlier detection by sampling with accuracy guarantees. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 767–772. ACM.

Wu, R., Wang, B., Wang, W., and Yu, Y. (2015). Harvesting discriminative meta objects with deep cnn features for scene classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1287–1295.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xu, W. and Tan, Y. (2019). Semisupervised text classification by variational autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14. To appear.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2017). Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2008–2026.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. B. (2018). Noisy natural gradient as variational inference. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5847–5856.

Zhao, S., Song, J., and Ermon, S. (2017). Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929.

Zimek, A., Schubert, E., and Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387.

université PARIS-SACLAY

ECOLE DOCTORALE N°580

Sciences et technologies de l'information et de la communication (STIC)

Titre: Apprentissage pour la détection d'anomalies, avec application au contrôle de qualité d'acquisition de données de l'expérience Compact Muon Solenoid

Mots clés: Détection d'anomalie, inférence variationnelle, contrôle de qualité, physique des hautes énergies

Résumé: La surveillance de la qualité des données qui proviennent des expériences de physique des hautes énergies est une tâche exigeante mais cruciale pour assurer que les analyses physiques sont basées en données de la meilleure qualité possible. Lors de l'expérience Compact Muon Solenoid opérant au Grand collisionneur de hadrons du CERN, le paradigme actuel d'évaluation de la qualité des données est basé sur l'examen détaillé d'un grand nombre de tests statistiques. Cependant, la complexité toujours croissante des détecteurs et le volume des données de surveillance appellent un changement de paradigme. Ici, les techniques de Machine Learning promettent une percée. Cette thèse traite du problème de l'automatisation applique à la surveillance de la qualité des données avec les méthodes de détection des anomalies d'apprentissage automatique. La grande dimensionnalité des données empêche l'utilisation de méthodes de détection classiques, pointant vers de nouvelles, basées sur l'apprentissage en profondeur. Les anomalies causées par un dysfonctionnement du détecteur sont difficiles à énumérer *a priori* et rares, ce qui limite la quantité de données étiquetées. Ainsi, cette thèse explore le paysage des algorithmes existants avec une attention particulière aux problèmes semi-supervisés et démontre leur validité et leur utilité sur des cas de test réels en utilisant les données de l'expérience. Dans le cadre de ce projet, l'infrastructure de surveillance a été encore optimisée et étendue, offrant des méthodes plus sensibles aux différents modes de défaillance.

Title: Machine Learning Anomaly Detection Applications to Compact Muon Solenoid Data Quality Monitoring

Keywords: Anomaly detection, variational inference, quality control, High Energy Physics

Abstract: The Data Quality Monitoring of High Energy Physics experiments is a crucial and demanding task to deliver high-quality data used for physics analysis. At the Compact Muon Solenoid experiment operating at the CERN Large Hadron Collider, the current quality assessment paradigm, is based on the scrutiny of a large number of statistical tests. However, the ever increasing detector complexity and the volume of monitoring data call for a growing paradigm shift. Here, Machine Learning techniques promise a breakthrough. This dissertation deals with the problem of automating Data Quality Monitoring scrutiny with Machine Learning Anomaly Detection methods. The high-dimensionality of the data precludes the usage of classic detection methods, pointing to novel ones, based on deep learning. Anomalies caused by detector malfunctioning are difficult to enumerate *a priori* and rare, limiting the amount of labeled data. This thesis explores the landscape of existing algorithms with particular attention to semi-supervised problems and demonstrates their validity and usefulness on real test cases using the experiment data. As part of this project, the monitoring infrastructure was further optimized and extended, delivering methods with higher sensitivity to various failure modes.

Université Paris-Saclay
CNRS, Laboratoire de recherche en informatique
91405, Orsay, France