



PERSPECTIVE • OPEN ACCESS

Perspective on machine learning for real-time analysis at the Large Hadron Collider experiments ALICE, ATLAS, CMS and LHCb

To cite this article: S Astrand *et al* 2026 *Mach. Learn.: Sci. Technol.* **7** 013001

View the [article online](#) for updates and enhancements.

You may also like

- [Quantum mechanics and neural networks](#)
Christian Ferko and James Halverson
- [Quantum vs. classical: a comprehensive benchmark study for predicting time series with variational quantum machine learning](#)
Tobias Fellner, David A Kreplin, Samuel Tovey et al.
- [Learnable stability-aware computational grid coarsening for accelerating physics simulations](#)
Alexander Ryabov, Viacheslav Naumov, Sergei Shumilin et al.



PERSPECTIVE

OPEN ACCESS

RECEIVED
10 September 2025REVISED
12 November 2025ACCEPTED FOR PUBLICATION
8 January 2026PUBLISHED
20 January 2026

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Perspective on machine learning for real-time analysis at the Large Hadron Collider experiments ALICE, ATLAS, CMS and LHCb

S Astrand¹ , L Boggia², M Borsato^{3,4}, L Bozianu^{5,*} , C E Cocha Toapaxi⁶, F I Giasemis^{2,7} , J Hansen¹, P Inkaew⁸, K E Iversen¹ , P Jawahar⁹ , H Pineiro Monteagudo^{10,11} , M Olocco¹² and S Schramm⁵

- ¹ Lund University, Lund, Sweden
 - ² LPNHE, Sorbonne Université, CNRS/IN2P3, Paris, France
 - ³ Università Degli Studi di Milano-Bicocca, Milano, Italy
 - ⁴ INFN Sezione di Milano-Bicocca, Milano, Italy
 - ⁵ Université de Genève, Genève, Switzerland
 - ⁶ Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany
 - ⁷ LIP6, Sorbonne Université, CNRS, Paris, France
 - ⁸ University of Helsinki, Helsinki, Finland
 - ⁹ University of Manchester, Manchester, United Kingdom
 - ¹⁰ Università di Bologna, Bologna, Italy
 - ¹¹ Verizon Connect Research, Florence, Italy
 - ¹² Technische Universität Dortmund, Dortmund, Germany
- * Author to whom any correspondence should be addressed.

E-mail: leon.bozianu@cern.ch

Keywords: machine learning, real-time analysis, LHC experiments

Abstract

The field of high energy physics (HEPs) has seen a marked increase in the use of machine learning (ML) techniques in recent years. The proliferation of applications has revolutionised many aspects of the data processing pipeline at collider experiments including the Large Hadron Collider (LHC). In this whitepaper, we discuss the increasingly crucial role that ML plays in real-time analysis (RTA) at the LHC, namely in the context of the unique challenges posed by the trigger systems of the large LHC experiments. We describe a small selection of the ML applications in use at the large LHC experiments to demonstrate the breadth of use-cases. We continue by emphasising the importance of collaboration and engagement between the HEP community and industry, highlighting commonalities and synergies between the two. The mutual benefits are showcased in several interdisciplinary examples of RTA from industrial contexts. This whitepaper, compiled by the SMARTHEP network, does not provide an exhaustive review of ML at the LHC but rather offers a high-level overview of specific real-time use cases.

1. Introduction

The extraordinarily large datasets collected by the four Large Hadron Collider (LHC) experiments—ALICE, ATLAS, CMS and LHCb—since the start of LHC data-taking in 2009 represents the most precisely curated set of high-energy hadronic collisions ever recorded, but also an unparalleled technological challenge. Collisions at the LHC occur at a frequency of up to 30 MHz, producing data at a rate of one petabyte per second in the current LHC Run 3 data taking period. Managing this huge data rate requires real-time processing to determine what fraction is worth keeping, necessitating complex trigger and data acquisition systems.

This poses several critical issues unique to the LHC: the read-out of all sub-detector components within strict timing constraints, the reconstruction of physically motivated particle candidates and the storage of detector information from events of interest.

The trigger systems of the large LHC experiments play a pivotal role in their overall physics programme by dictating which events to keep or discard based only on the detector signatures of particles

produced in the final state of inelastic proton–proton collisions. The physics prospects of the LHC experiments will be limited by the latency and bandwidth constraints of their trigger systems. Whilst machine learning (ML) has a long history of usage in the field of high energy physics (HEPs), recent advances in computing have unlocked previously infeasible applications in resource constrained, real-time environments. Tasks such as physics object reconstruction or detector alignment and calibration can now be performed by modern ML models that are capable of interpreting the complex and high-dimensional detector data within strict latency requirements. The LHC collaborations have heavily relied on ML in real-time environments for analyses involving B-hadron final states, including landmark results such as the first observations of di-Higgs decays to final states involving bottom quarks. However, ML usage in real-time is continuing to expand, and dramatically increasing the sensitivity to physics results in numerous different ways.

The collaboration between academia and industry on the development of new tools dedicated to real-time ML applications has been crucial in improving their viability for the extreme conditions at the LHC. Moreover, many of the tools developed for the LHC triggers can also be used in corresponding real-time regimes in industry. In a number of cases, including LHC triggers, dedicated hardware is the only way to achieve the required inference latency.

The structure of the document is as follows: section 2 gives a brief overview of the four large LHC experiments and the structure of their detectors. Section 3 defines the real-time environment in the context of collider experiments and discusses the motivation and challenges associated with deploying ML there. Section 4 focuses on a number of representative examples of ML applications operating in real-time at the different large LHC experiments. Section 5 introduces commonalities between the HEPs field and industry and presents examples of cross-over or collaboration.

2. The large LHC experiments

The detectors of the four large LHC experiments are constructed with the goal of identifying final state particles whilst maintaining sensitivity to unknown particles not predicted by the standard model (SM). Many of the diverse technologies and complex instruments used by the experiments have been updated in the preceding long shutdown periods. In this section we briefly introduce the sub-detectors that make up each of the LHC experiments.

2.1. The ALICE detector

The primary goal of the ALICE experiment [1] at the LHC is to investigate ultrarelativistic heavy-ion collisions amongst others to better measure the properties of the quark-gluon plasma, a deconfined state of quarks and gluons theorised to exist in the early Universe [2]. The ALICE physics programme is centred on the study of lead nucleus collisions (Pb–Pb), proton-lead (p–Pb) collisions as well as proton–proton interactions.

The ALICE detector is composed of a number of detector subsystems [3]. The central barrel surrounding the central interaction point consists of a large solenoid magnet that generates a uniform magnetic field of up to 0.5 T along the beam direction. Inside the magnet, a set of detectors is located surrounding the beam axis radially with dedicated detectors used for tracking and particle identification. Immediately surrounding the beam pipe is the high-precision silicon tracker. The main tracking device is the time projection chamber (TPC). The transition radiation detector and time of flight subsystems provide accurate particle identification. Additional detectors, such as the muon spectrometer (MS), are located outside the central barrel in the forward beam direction.

Ahead of LHC Run 3 the ALICE trigger system was upgraded to a streaming readout system where collision data is continuously processed and recorded [4]. The central trigger system handles the data flow and synchronisation of sub-detectors before sending aggregated event information to the software-based high-level trigger (HLT). The HLT reconstructs the collision events, calibrates and aligns the different sub-detectors and performs data reduction.

2.2. The ATLAS detector

The ATLAS detector [5, 6] covers nearly the entire solid angle around the collision point. It consists of an inner tracking detector surrounded by a superconducting solenoid, electromagnetic and hadronic calorimeters, and a MS incorporating three large superconducting air-core toroidal magnets. The inner detector system is immersed in a 2 T axial magnetic field and provides charged-particle tracking in the pseudorapidity range $|\eta| < 2.5$. The high-granularity silicon pixel detector (PIX) covers the vertex region and typically provides four measurements per track, the first hit normally being in the insertable B-layer. It is followed by the semiconductor tracker and the transition radiation tracker.

The ATLAS calorimeter system covers the pseudorapidity range $|\eta| < 4.9$. Within the region $|\eta| < 3.2$, electromagnetic calorimetry is provided by barrel and endcap high-granularity lead/liquid-argon (LAr) calorimeters. Hadronic calorimetry is provided by the steel/scintillator-tile calorimeter, segmented into three barrel structures within $|\eta| < 1.7$, and two copper/LAr hadronic endcap calorimeters. The solid angle coverage is completed with forward copper/LAr and tungsten/LAr calorimeter modules optimised for electromagnetic and hadronic energy measurements, respectively. The MS comprises separate trigger and high-precision tracking chambers measuring the deflection of muons in a magnetic field generated by the superconducting air-core toroidal magnets.

2.3. The CMS detector

The central feature of the CMS apparatus is a superconducting solenoid of 6 m internal diameter, providing a magnetic field of 3.8 T [7, 8]. Within the solenoid volume are a silicon pixel and strip tracker, a lead tungstate crystal electromagnetic calorimeter (ECAL), and a brass and scintillator hadron calorimeter (HCAL), each composed of a barrel and two endcap sections. Forward calorimeters extend the pseudorapidity coverage provided by the barrel and endcap detectors. Muons are measured in gaseous ionisation detectors embedded in the steel flux-return yoke outside the solenoid. At the start of 2017, a new PIX was installed [9] to provide four-hit pixel coverage in the pseudorapidity range $|\eta| < 2.5$.

2.4. The LHCb detector

The LHCb experiment [10] is a dedicated heavy-flavour experiment designed to study the properties of b - and c -hadron decays including those involving CP and flavour violation. Since its inception, however, the LHCb physics programme has grown substantially to include a large and diverse set of measurements, including studies of electroweak physics and searches for dark matter candidates.

The LHCb detector is a single-arm forward spectrometer instrumented to achieve acceptance in the pseudorapidity range, $2 < \eta < 5$ described in detail in [11]. The LHCb tracking system consists of the vertex detector (VELO), the upstream tracker and the scintillating fibre tracker (SciFi). Placed immediately after the VELO and the SciFi tracker are two ring imaging Cherenkov detectors (RICH). The ECAL and HCAL calorimeters are placed after the RICH detector. Finally, the five muon stations lie furthest from the interaction point. Aside from its non-hermetic geometry the LHCb detector exhibits a number of other differences with respect to the other LHC experiments. The VELO used for tracking charged particles is located only 5.1 mm from the beam. This proximity provides excellent impact-parameter resolution, allowing efficient separation of displaced tracks from heavy-flavour decays from the large background of tracks originating at the primary vertex. During the ongoing LHC Run 3 the operating luminosity of the LHCb experiment is levelled to a rate of $2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$. At this luminosity, there are approximately 6 proton–proton interactions per bunch crossing—a factor of five more than in LHC Runs 1 and 2 [12].

The LHCb trigger system was upgraded ahead of LHC Run 3 and has entirely removed the hardware-based trigger step and replaced it with a second software-based trigger. The upgraded software-based trigger steps, HLT1 and HLT2, process an increased event readout using a heterogeneous computing architecture based on GPUs [13].

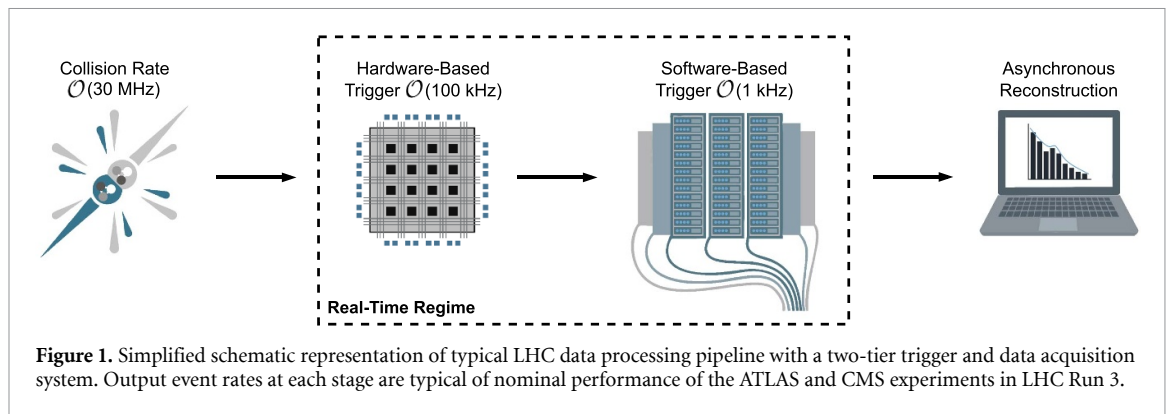
3. Real-time analysis at collider experiments

3.1. The real-time environment

The data analysis workflows of the four LHC experiments generally adhere to two-stage processing pipelines. First, events are processed synchronously, or *online*, within each experiment's dedicated trigger system before they are stored. Then, the second data processing step occurs asynchronously, or *offline*, across the Worldwide LHC Computing Grid [14, 15]. The reconstruction, selection, calibration and monitoring performed online can be considered to occur in 'real-time', where power, latency and radiation constraints determine the practicability of different algorithms. Figure 1 shows a simplified representation of this pipeline.

The development and performance of the triggers employed by the different LHC experiments is beyond the scope of this review paper, but can be found in [16–20] as well as in a high-level comparison in [21, 22].

In the real-time regime, specifically the hardware-based trigger (the first of the two stages), it is typically not possible to read out the data from the entire detector for every event at the 30 MHz collision rate. Instead, it is necessary to only read-out information in selected regions of interest or sub-detectors that meet the strict latency requirements. The consequence of the limited and coarse information at the trigger level is a loss in the accuracy or resolution of the online algorithms which rely on lower-quality,



truncated or incomplete inputs. In the ATLAS and CMS experiments the result of the hardware-based trigger is the rejection of approximately 99.75% of the collisions, those that are accepted proceed to the second, software-based, stage of the trigger.

In the software-based trigger, more complex methods and algorithms can be deployed. With a reduced throughput, the latency requirements are relaxed with respect to the hardware-based trigger. At this stage, it is also possible to read out information from all sub-detector systems and in their full granularity. Typically, the software trigger will reconstruct the physics objects that have already been estimated by the hardware trigger using more refined methods, including ML, utilising the information from the entire detector. Final trigger decisions are then made using these physics objects.

In the triggers of the LHC experiments the online reconstruction performance aims to match the offline performance as closely as possible (in some cases even being used offline directly). Often, this is not possible due to the limitations imposed by the immense volume of data produced by the LHC and the reduced fidelity of the detector readout in real-time in comparison to offline. However, due to increases in CPU performance, algorithmic efficiency and the advancement of ML models the online reconstruction performance has greatly improved in the most recent data taking period. As a consequence, the performance of online reconstruction algorithms now frequently approaches that of their offline counterparts.

The usage of ML at collider experiments has a long history [23–27] from simple multivariate techniques to more complex models. The use of ML has become more widespread in recent years due to the applicability of ML models in the ‘big data’ domain at the LHC where copious collision events and Monte Carlo (MC) simulations provide immense amounts of training data. The use cases for ML in HEP have proliferated beyond enhancing signal identification and now include physics reconstruction, calibration, simulation generation, detector alignment, anomaly detection and data compression.

3.2. Integration of ML into existing workflows

The design of the LHC trigger systems undergoes near constant development. In recent years, considerable advancements have been made in the capabilities of online data processing expanding the potential frontier of real-time data processing. In order to maximise the physics potential of the prodigious dataset produced in LHC Run 3, and beyond, the online reconstruction must handle the uniquely challenging data processing conditions. This necessitates the continuous development of new reconstruction techniques or the adaptation of existing offline algorithms to comply with the requirements of online deployment. This cyclical developmental cycle has proven to be effective in bringing offline-like performance to the real-time regime at the large LHC experiments. The progress in online performance achieved in recent years has been ameliorated by the emergence of more sophisticated ML models and dedicated hardware accelerators.

The potential applications of ML methods within real-time data processing pipelines are diverse, spanning from relatively simple models deployed on dedicated hardware to more complex physics object reconstruction with modern neural network (NN) architectures. Furthermore, ML models can offer quick approximations to more complex, performant algorithms. The precise placement of such models in the dataflow is dependent on the different computational constraints and detector read-out that is possible within the wider trigger sequence.

3.2.1. Machine learning on dedicated accelerators

An important building block of the hardware triggers used at the LHC experiments is a class of integrated circuits called field programmable gate arrays (FPGAs) [28] that are used to perform low-level

operations using logic gates, digital signal processors, flip-flops and look up tables. Unlike application-specific integrated circuits, FPGAs can be reconfigured after their initial manufacturing to perform modified or entirely new functions. FPGAs have been used extensively at the LHC experiments, particularly in the hardware-based triggers, where specialised hardware has been required to meet the throughput and latency specifications.

In recent years, FPGAs have evolved from specialised processors to more general hardware accelerators. One major benefactor has been the ML community—a number of tools have been developed to port ML models onto individual FPGAs. Crucially, the FPGA is able to parallelise many of the operations used inside ML models including large matrix multiplications. This has enabled significant growth in the deployment of more complex ML models onto FPGAs in the context of HEP triggers. The interplay between industry and academia has produced a number of open-source projects for deploying ML models on dedicated hardware including Conifer [29], FINN [30], FwXMachina [31], HLS4ML [32, 33] as well as proprietary platforms developed by vendors providing tools for specific devices such as Vitis AI [34] and Intel AI Suite [35].

These tools allow developers to interact with the FPGA board at an abstract level above hardware description languages (HDLs) and act as a translator between software programming languages, and low-level HDL including Verilog and VHSIC HDL. These developments have removed one potential barrier to ML deployment and therefore enable a faster development cycle.

Associated developments in the training of ML models before deployment to FPGAs have seen vast improvements in model performance in resource constrained environments. Examples of these developments include quantisation-aware training (QAT) [36], knowledge distillation [37–39] and pruning [40]. These techniques allow models to be trimmed, evaluated at lower numerical precision or to approximate the performance of larger models whilst incurring only a small degradation in performance [41, 42]. The available tools have become more widespread and are well-supported by the community at large. A number of frameworks including qKeras [43, 44], Brevitas [45] and ONNX Runtime [46] all facilitate QAT or the mixed-precision conversion of existing ML models.

As the suite of tools for deploying ML on hardware matures and more functionality is added their usage will increase and the capabilities of the FPGAs will improve, thus allowing ML models to be brought closer to the front-end electronics of the LHC detectors in the real-time environment of the hardware-based triggers.

3.2.2. ML models and software

The evolution and development of ML libraries and frameworks has been characterised by the interplay between large technology companies and the open-source community. These libraries essentially provide a layer of abstraction between ML developers and the linear algebraic operations necessary for particular ML architectures. The LHC experiments, and the wider HEP community, are both consumers of and contributors to these efforts including though not limited to Caffe [47], PyTorch [48], Scikit-Learn [49], Tensorflow [50] and TMVA in ROOT [51].

The proliferation of ML expertise within HEP and the large number of user-friendly libraries have reduced the barriers of entry for the LHC experiments to examine more complex ML methods and reduced the amount of duplicated work related to MLOps [52].

The data-intensive nature of the experiments open themselves up to approaches that can be refined with increased training data. In the real-time regime, a key determinant of the impact of ML will be the integration with the existing experiment codebases. The trigger software of each of the LHC experiments is tailor-made for the current experimental configuration. It is noteworthy that the interface between the present trigger software stacks and evolving, external libraries continues to be crucial to the success of ML deployment.

3.3. Associated challenges

The use of ML in a real-time environment provides an effective and powerful method of data analysis, in some cases alleviating the computational cost of legacy or iterative algorithms. In order to achieve the required latency constraints, it is often necessary to fit the ML architecture, including all trained weights, on-chip or within the wider trigger computing network. On specialised hardware, ML models must also accommodate other essential services including those for data input and output. As previously mentioned, the latency and throughput requirements can limit the maximum complexity of the models. In spite of this, model performance in the real-time environment cannot be guaranteed to match that of offline algorithms. There exists a trade-off between the possible latency or throughput and the accuracy of ML models deployed in resource constrained environments. Furthermore, ML models intended

for online deployment undergo significant testing in order to reduce the potential for data to be irretrievably lost. Typically, ML algorithms are validated in the real-time environment by shadowing existing algorithms during online data taking whilst the effects on trigger performance and data taking efficiency are assessed.

The use of ML models can often reduce the need for physically motivated engineered features, rather than relying on data-driven insights that are derived from the correlation of input features. The non-parametric nature of some ML models means that it can be difficult to interpret the model output for a given set of inputs. This can obscure the underlying physics models and processes that govern the collisions measured at the LHC and elsewhere. One recent direction of research in HEP has been in encoding domain-specific inductive biases from HEP within ML models directly. This includes enforcing a monotonic response to input features, equivariance under some transformation of the inputs [53, 54], or similar physics-informed requirements and representations [55, 56].

Related to the difficulties of interpretation are the susceptibility of many ML models to out-of-distribution effects or edge cases. Deployment in the data analysis workflow at the LHC experiments can lead to unforeseen failure modes of the model due to the large statistics involved and the potential domain shift from model training to inference. Therefore, it is crucial that the model outputs are verifiable and interpretable such that unexpected behaviour can be both identified and rectified. The ability to understand the inner workings of ML models is essential both to understand the underlying physics and to strengthen the certainty with which claims can be made *a posteriori*. The reliability of a particular model takes on increased importance at collider experiments, where final states of inelastic collisions are governed by quantum field theory and can only be inferred statistically. The interpretability and explainability of ML models is an active area of research, particularly in the HEP field [57, 58].

Another challenge that accompanies the use of ML in the HEP domain is the inherent reliance on MC simulation. The supervised training of large ML models is predicated on the existence of large datasets with reliable ‘truth’ labels that are used to discover the relationship between different input variables and the desired output. Collider experiments have used MC simulation for a variety of applications outside of ML training, but the advent of larger and more complex ML models and their corresponding increase in performance has meant that ML performance is now reliant on the MC generators used to produce the simulation. It is possible that sophisticated ML models are able to find or recognise relationships that exist in simulation but do not appear in real particle collisions, therefore degrading the model performance in its application to real data. This effect would manifest as a systematic deviation related to the mismodelling of the simulation. Though this problem is an open area of study, the large LHC experiments employ several strategies to mitigate it, including: validation using adjacent control regions, weakly supervised learning using real data and parameterising known nuisance parameters within the network architecture [59].

The challenges detailed above are exacerbated in the real-time regime due to the nature of the trigger and data acquisition systems at the LHC experiments; errors in event selection are unrecoverable since particle collisions are either saved to storage or discarded. In this way, there is no way to compensate for errors during data taking periods—if an ML model has some inaccuracy or bias due to it is MC simulation labelled training corpus or an unexpected edge case then entire classes of collision events may be irretrievably rejected.

4. Examples of ML for real-time analysis (RTA)

4.1. Physics reconstruction

4.1.1. Track Reconstruction in LHCb

At the LHCb experiment, the reconstruction of charged particles in the tracking system plays a crucial role in its physics programme. Ahead of LHC Run 3, the LHCb trigger system was migrated to a fully GPU-based first-level trigger (the Allen system [13]) that reduces the initial dataflow throughput from 5 Tb s^{-1} to less than 200 Gb s^{-1} [60]. The use of GPUs at this stage of the real-time data processing pipeline opens up the potential for deploying complex ML-based algorithms to aid in the reconstruction and classification of events. One example of this is the ongoing development of a graph NN (GNN) to identify tracks in the VELO sub-detector.

In LHC Run 3, there are approximately 5 proton–proton collisions per bunch recorded by LHCb in the VELO sub-detector acceptance. The GNN reconstruction, called ETX4VELO [61], implements track finding in five steps: hit graph construction, edge classification, edge graph construction, triplet classification and track construction. The different steps improve the efficacy of the ETX4VELO algorithm for tracks with challenging experimental signatures, including intersecting tracks or shared hits, skipped VELO planes, multiple hits per plane and tracks containing electron–positron pairs. Functionally, the

algorithm can be broken down into several sequential models: an embedding DNN, an edge classification GNN with multiple rounds of message passing and a weakly connected component step where the final tracks are retrieved.

The performance of ETX4VELO rivals the track reconstruction used in the Allen framework across all track types, with particularly pronounced improvement for electron tracks. Moreover, the number of fake tracks produced is reduced to less than 1%, compared to previously more than 2%. Critically, ETX4VELO detects 15% more electrons that only leave tracks in the VELO. Work is ongoing to explore the deployment of ETX4VELO on dedicated FPGA accelerators [62] using the HLS4ML library. Such an extension could provide further improvements including enhanced computational performance, greater power efficiency and higher throughput while more easily accommodating the requirements of future HEP experiments.

4.1.2. Electron and photon identification in ATLAS

In the ATLAS experiment trigger system, there are several fast algorithms employed as preselections to quickly reject dominant background sources before more complex and performant algorithms are run downstream to produce the final event selections [63]. One example of an ML-based preselection is the Ringer algorithm [64], which is used to identify electrons and photons in the ATLAS calorimeter.

The Ringer is a DNN that takes as input the energy sum of cells in $\mathcal{O}(100)$ concentric rings centred on the cell with the highest energy in each calorimeter sampling layer. The ring inputs exploit the approximately conical structure of electromagnetic showers and their lateral development within a single calorimeter layer. Further, the concentric rings are statistically dependent both laterally and longitudinally which allows the DNN to leverage their correlations. The concatenated vector containing the normalised energy sum of each concentric ring is fed into an ensemble of simple MLPs defined for each $E_T \times \eta$ region in the calorimeter.

The Ringer algorithm deployed in the ATLAS software trigger was marginally slower than the previous cut-based approach. Despite this, its greater rejection efficiency resulted in a 50% reduction in the overall CPU demand for single electron triggers.

4.1.3. Tau reconstruction in CMS

The tau lepton, with its large mass and short lifetime, typically decays before reaching the active material of LHC experiments. It is therefore necessary to reconstruct this lepton via its decay products alone. Two-thirds of tau lepton decays result in hadrons and a single tau neutrino hence, their identification partially relies on standard jet reconstruction. In its LHC Run 3 trigger system, the CMS experiment utilises a custom DNN incorporating convolutional layers to identify hadronically decaying tau candidates [65]. This method is able to simultaneously reject the quark or gluon initiated jets, electron and muon backgrounds that contaminate the hadronic tau signature. Previously, separate discriminators were used to reject jets, electrons and muons respectively.

The DNN architecture, called DeepTau [66], utilises a combination of high-level input features as well as local calorimeter information to discriminate the hadronic tau candidates from all relevant backgrounds at once. The DeepTau network consists of three subnetworks that each handle a different data input: the outer cells, inner cells and high-level variables. The subnetwork outputs are concatenated and passed through four hidden layers before yielding a final output vector that can be interpreted as the probability that the object belongs to each class $\{\tau, \text{jet}, \mu, e\}$. The subnetworks processing the inner and outer cells are based on convolutional layers that act on embedded features derived from the grid-like cell information. The convolutional layers successively reduce the grid size before it is concatenated and passed to the output dense layers. The 47 high-level variables pass directly through 3 dense layers before concatenation.

The DeepTau network achieves an improvement in efficiency of between 10% and 30% for a given jet rejection, while the efficiency of tau candidates passing the loose identification threshold increases by 14% with DeepTau compared to the baseline performance.

4.1.4. Jet flavour tagging in ATLAS

The identification of jets initiated by heavy-flavour hadrons (b - and c -hadrons) is a fundamental component of the physics programmes of the different LHC experiments. Efficient discrimination between heavy-flavour jets and those arising solely from light quarks or gluons is essential to suppress the overwhelming QCD dijet background in proton–proton collisions. Jets may also contain decay products of electroweak bosons (e.g. W , Z), top quarks, τ leptons or even the Higgs boson. Experiments around the LHC apply algorithms to identify or ‘tag’ these jets using their distinguishing physical properties to separate them from light-flavour hadrons or gluon-induced jets.

In LHC Run 3 the ATLAS experiment has introduced new ML models to replace existing flavour tagging algorithms that previously relied on high-level observables engineered specifically for jet flavour classification. The new ML models utilise low-level features including parameters related to the primary vertex and the tracks associated to each jet to discriminate between the different jet flavours. The collection of low-level features implicitly contains information about the hadron lifetime and the secondary vertex mass.

The models now used in the ATLAS trigger are based on either Deep Set (fastDIPS [67]) or Transformer (GN1 [68] and GN2 [69]) architectures. These network architectures directly combine the track and jet information to infer the jet flavour. Both architectures embed the the input features before combining them via a summation or attention mechanism respectively. The model output is a multi-class probability score across the different jet flavours that is combined into a single discriminant that is used to make a final determination. The fastDIPS and transformer-based models are deployed in the software-based trigger in ATLAS. In the real-time regime the model performance on jets with low transverse momenta is particularly important because of the relatively large cross-section of multijet production at the LHC. Therefore the flavour-tagging algorithms can be used to reduce the light-jet background and thus the event rate to a feasible level. Moreover, for some signatures these ML models can be used as an approximate preselection applied earlier in the software-based trigger to provide a preliminary reduction in CPU cost. The models can be executed using coarse or unrefined input features and still provide an acceptable level of rejection that prevents the execution of more computationally expensive methods downstream.

4.1.5. Pile-up mitigation

At the luminosity at which the LHC collides bunches of protons it is likely that multiple proton–proton interactions occur concurrently in the same bunch crossing. The additional proton–proton interactions that occur alongside the most energetic primary vertex (referred to as the hard scatter vertex) are considered ‘pile-up’ interactions. The average number of simultaneous collisions in the ATLAS and CMS experiments is estimated to exceed 50 in LHC Run 3 [70].

The goal of the LHC experiments is, in general, to analyse only the hard scatter vertex from each bunch crossing, in order to most benefit from the operational centre of mass energy of the LHC. The pile-up interactions deposit energy in each of the detectors and intermix with the signals produced by the interaction of interest. Most of these pile-up interactions are soft inelastic collisions that produce relatively low-energy deposits with an isotropic distribution. In order to be able to accurately distinguish the most energetic interaction in each bunch crossing, the pile-up contributions must be suppressed. This suppression is crucial in a real-time environment where a decision is made about whether to store an event or not based on the candidate physics objects present in the event. Particles originating from pile-up interactions can erroneously allow events to pass trigger hypotheses—thus contaminating the final event selection. It is critical to the operation of LHC triggers that the additional pile-up particles are identified, if possible, and accounted for in real-time.

One approach, PUMML [71], aims to use CNNs to remove the pile-up contamination from the interaction of interest. The PUMML algorithm targets the transverse momenta of neutral particles originating from the primary vertex. Neutral particles represent a challenging signature to reconstruct as there is no corresponding track pointing back to the primary vertex. The network input is a 36×36 pixel image centred on a single jet axis with three channels. Each channel corresponds to a different source of transverse momentum (primary vertex charged, pile-up vertex charged and total neutral). When evaluated on simulation the PUMML algorithm matches the efficiency of current rule-based algorithms at removing neutral pile-up particle transverse momenta.

Another, more recent application of ML to pile-up mitigation is PUMA [72], which uses a sparse transformer to model pile-up contributions. PUMA reconstructs the hard energy fraction of each reconstructed particle candidate in an event such that it may be corrected to reduce pile-up contamination. This problem formulation accommodates the finite spatial and energy resolution of the detectors and that it may not be possible to fully separate contributions from the primary or pile-up vertices. PUMA uses a sparse self-attention mechanism to reduce the time and memory complexity of full self-attention, since the number of pile-up particles can exceed the constraints of even large GPUs. The PUMA model outperforms classical pile-up suppression methods across the board with a 20% improvement in the resolution of missing transverse momentum per event.

4.1.6. Identifying heavy-flavour decays in LHCb

In LHC Run 3, the LHCb topological trigger was updated to include a monotonic Lipschitz NN [73, 74] as a final selection criterion in the identification of displaced b -hadron decays. After an initial cut-based

preselection is used to discard prompt and soft background candidates the monotonic Lipschitz NN is used to make an additional selection. Two identical models are trained separately, targeting two- and three-body decays respectively, which are defined based on the multiplicity of charged particles reconstructed in the decay products of the b -hadron secondary vertex.

The networks take as input a vector of features derived from the final-state tracks of the multi-body decay products that capture the overall topology and kinematics of the system. Observables include the candidate transverse momentum, decay-vertex fit quality, flight distance, distance of closest approach of charged tracks, multi-body corrected mass value [75] and the impact parameter with respect to the primary vertex. The NN enforces a constraint on the maximum value of the Lipschitz constant of the learnt decision-boundary function. The bounding of the Lipschitz constant ensures that events that are in close proximity in the feature input space are also close in the final decision space of the network. By setting a strict upper limit on the classifier response variation, the magnitude of the effects of finite detector resolution and other experimental instabilities are limited.

The monotonic Lipschitz architecture offers improved robustness under varying detector conditions and contains physically motivated inductive biases. The performance of the ML model is improved with respect to the previously employed BDTs used in LHC Run 2, whilst also satisfying the increased bandwidth and latency requirements of the improved LHCb trigger in LHC Run 3.

4.2. Object calibration

4.2.1. TPC calibration in ALICE

Before LHC Run 3 the ALICE trigger and data acquisition system was upgraded to a streaming readout—data is continuously processed and recorded for every collision [76]. This novel readout system represents a paradigm shift in data management at an LHC experiment and presents a number of challenges and opportunities for its future operation. Among the challenges, the real-time calibration of detector signatures will be of paramount importance in order to ensure optimal detector operation and rapid feedback in case of detector malfunction.

The TPC is the main detector for tracking and particle identification in ALICE. The TPC identifies charged particles directly by reconstructing their trajectories as they pass through the TPC volume. The signals are amplified by gas electron multipliers. The space-charge density within the TPC depends on the number of pile-up ions in one ion drift time. The expected distribution of space-charge density at 50 kHz of lead-lead interactions can be derived, investigated and tested using MC simulations.

The calibration of the TPC aims to correct the average space-charge distortions and the distortion fluctuations during both synchronous and asynchronous reconstruction. The timing constraints for the distortion fluctuation calibrations during readout ($\mathcal{O}(10\text{ ms})$) can be met whilst using a fast ML model. A CNN model has been developed to predict the space-charge distortion fluctuation corrections [77] based on a U-net architecture [78], where the input image is a 3-dimensional representation of the space-charge distortions fluctuations map in a $90 \times 17 \times 17$ or $180 \times 33 \times 33$ grid-like representation. Development of the CNN is ongoing though initial performance appears to depend on the distance from the fiducial boundaries of the TPC.

4.3. Anomaly detection (AD)

AD in HEP can be described as the process of identifying out-of-distribution events that broadly exhibit deviations from the SM with no *a priori* assumption on the specific alternative hypothesis. This includes both the detection of erroneous events affected by stochastic noise inherent to the LHC subdetectors as well as rare events, potentially containing physics beyond the SM (BSM).

These two branches of AD, while correlated, address different classification problems. The first classifies events that have been impaired as a result of noise or other undesirable detector effects, ideally at individual component levels. To maintain a high data taking efficiency and prevent the contamination of datasets, these events must be monitored and, if necessary, removed. This process is broadly referred to as data quality monitoring (DQM). In HEP, noisy events are considered anomalous and standard detector performance is presumed to make up the bulk of the in-distribution events. The identification of these anomalous noisy events not only aids in cleaning the datasets collected by the detector, but also helps in understanding novel detector effects, such as noise bursts or high-voltage trips that can occur during data taking.

The second branch involves the identification of rare and unusual events, and is used as a tool in new physics searches by isolating events that exhibit behaviour inconsistent with the SM. These events can be saved for further analysis whilst remaining agnostic to the specific type of BSM physics model [79].

The operation of these AD methods occurs both online and offline in HEP experiments. However, the use and performance of AD models in real-time has proliferated in recent years as model capabilities and inference speed have improved. This has enabled more AD models to be deployed earlier in the event selection process at the LHC experiments, enabling less biased selections in model-agnostic searches for new physics and greater coverage with DQM earlier in the trigger system.

4.3.1. Data quality monitoring in ATLAS and CMS

The rate of collisions at the LHC during the Run 3 data taking can be up to 30 MHz, this represents an enormous flow of data that must be filtered, compressed and stored. Monitoring the data quality is therefore an essential task. The current DQM system in most LHC experiments is split into two categories: during data-taking (online) and asynchronously after data storage (offline). Most often both online and offline monitoring involve the comparison of histograms containing event-level variables to a predefined reference by a human expert. In the context of RTA, the online DQM is often dependent on a fraction of representative events written out to dedicated calibration or monitoring data streams that are promptly reconstructed and used to more quickly identify issues that could immediately affect data-taking efficiency.

The ATLAS and CMS experiments have both begun to examine the effectiveness of unsupervised ML models to assist human experts within DQM systems [80, 81]. A common example is the use of autoencoder (AE) architectures that are trained on examples of nominal detector performance to reconstruct the original input and therefore do not need truth labels. AE models typically consist of two NNs used in sequence to first ‘encode’ the input to some lower-dimension latent space and then to ‘decode’ the resultant latent vector representation back to the original input space. The performance of the AE compression-reconstruction is evaluated using a reconstruction loss which compares the model output to the original input.

In CMS a CNN-based AE model focussing on the ECAL was deployed online in the DQM workflow during 2022 in LHC Run 3 [82]. Occupancy histograms from the barrel and endcap sub-detectors are processed as two-dimensional images and a residual NN [83] is used to first compress the image into an informative low-dimensional representation before the decoder upsamples the latent representation back to the original image size. The mean squared error between the input and the AE reconstructed output is used to optimise the model’s performance on standard detector operation, as defined by experts. The addition of this AD model into the DQM has proven to be highly efficient with anomaly identification estimated to be approximately 99% in validation datasets with controlled noise injections. The addition of the CNN-based AE has complemented the existing DQM by simplifying decisions taken by detector experts and reducing the number of reported false alarms.

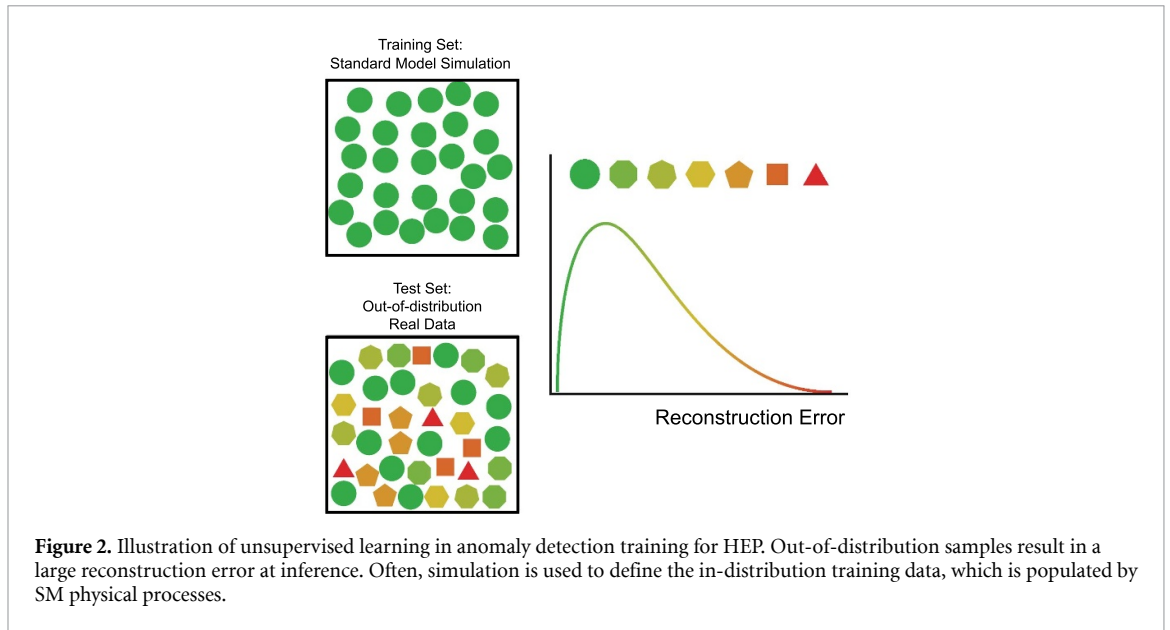
The ATLAS experiment has developed a different AE-based model for DQM [84], based on RNN blocks called long-short term memory [85] (LSTM) for both the encoding and decoding of the detector information.

In this application, data from the LAr calorimeters in the ATLAS detector are treated as multidimensional time-series where periods of nominal detector performance are punctuated by scarce, brief intervals of anomalous activity due to stochastic noise. The majority of noise bursts in the LAr calorimeter have a duration of less than 60 s.

The LSTM-based AE model classifies a time-series sample as standard detector performance or noisy in an unsupervised regime. The LSTM-based model is able to capture long-term dependencies and correlations within the data, in the context of DQM this is pertinent since the noise bursts in the LAr calorimeter corrupt data over contiguous periods of data-taking. The input data are comprised of normalised features related to clusters of calorimeter cells taken from different regions of the detector. The preliminary results from this work are promising and require few modifications to be run in real-time in the ATLAS trigger. Moreover, the performance was observed to extend to heavy ion collisions without retraining.

4.3.2. New physics searches in CMS

A fundamental assumption of AD in the context of searches for new physics is that any physics signature produced in the LHC detectors will be distinct from the overwhelming SM background present, represented pictorially in figure 2. Traditional physics analyses conducted at the large LHC experiments typically depend on particular signatures that arise from a predetermined set of signal hypotheses that can be simulated. AD-based BSM searches on the other hand intend to exploit the potential dissimilarities between new physics and SM processes to evaluate any BSM model in the relevant phase space. In this way, AD-based new physics searches can be used as a safeguard against biasing the direction of



searches in areas with theoretical motivation, thereby allowing under-investigated or unposed models to be probed further.

In addition, AD-based physics searches have benefited from the increased inference speed of ML models that allow AD scores to be calculated at the trigger-level. Therefore allowing the experiments to cast a wider net with reduced dependence on the standard set of triggers. The triggers in LHC experiments have been carefully optimised to increase both the sensitivity to their existing search programmes and the precision of their SM measurements. Deploying AD algorithms earlier in this chain allows more events up to the maximum collision rate of 30 MHz, to be examined before they are discarded by the model-dependent trigger and data acquisition algorithms.

The CMS experiment has developed and deployed two complementary AD networks to be placed in the initial, hardware-based trigger that is composed of custom hardware with a strict latency requirement of $4 \mu\text{s}$ [86]. Calorimeter Image Convolutional Anomaly Detection Algorithm (CICADA) [87] is a two-dimensional CNN that translates calorimeter region energy deposits into pixels in a 18×14 image. The training data is made up of zero-bias data events and since the network does not generalise well, by design, events that do not resemble the SM training set will have a higher reconstruction error. AXOL1TL [88] (Anomaly eXtraction Online Level-1 Trigger aLgorithm) is another ML model used by CMS to separate rare and exotic events in real-time. The AXOL1TL architecture is a Variational Autoencoder (VAE) that is also trained on zero-bias data and takes up to 19 coarsely-defined physics objects reconstructed in the L1 trigger as input. The loss term for the VAE includes a full regularisation term (KL divergence) as well as the MSE loss. Both models were trained using QAT and synthesised using the HLS4ML library discussed in section 3. Both algorithms were then each deployed on a single Xilinx Virtex-7 FPGA board. The CICADA model has an approximate latency of $\mathcal{O}(100)$ ns whereas for AXOL1TL the inference speed is just 50 ns.

The novel use of these models in the trigger demonstrates that model-independent methods for AD are possible in real-time applications, even in environments that are severely resource constrained and have extreme latency requirements.

5. Synergies between HEP and Industry

5.1. Common challenges of real-time analysis

Even in the era of big data, the computing challenges experienced at the LHC are unique. Proton–proton collisions occur with a frequency of up to 30 MHz in the LHC, corresponding to a streaming bandwidth of approximately 1 PB/s. This rate of data generation is among the largest in the world and, coupled with the imposed latency requirements, results in computing challenges without precedent. The trigger systems used in LHC experiments are subject to some of the largest global data throughputs whilst enforcing some of the shortest latency requirements.

As previously discussed, it is not possible for LHC experiments to record all events due to both throughput and storage limitations. Moreover, the overwhelming majority of LHC collisions produce

well-understood or uninteresting final state particles—it would not be worthwhile for LHC experiments to record these events at all. To accommodate the constraints of the readout and storage systems and advance the physics programme of the experiments, the trigger accepts on average only 1 in 30 000 collision events.

There are many examples from industry that also confront strict data throughput and latency requirements. Machine learning has become one of the most prominent tools used in real-time applications in industry in a diverse set of domains, including, though not limited to meteorological prediction and simulation, medical imaging, cybersecurity monitoring and natural language processing.

Industrial applications, similarly to the LHC, contend with a huge volume of data and must synthesise the vast amounts of information into a more manageable size in order to maximise the benefits of the scale of available data. The shared challenges between the HEP field and industry engender shared solutions to common problems, such as open source software frameworks, accessible tools for high-performance computing and cross-platform support among vendors of hardware accelerators.

5.2. Applications in industry: time-series anomaly detection

In industry, one real-time application of ML models is the detection of anomalies in time-series data. For example, IBM France is pursuing the development of time-series AD methods for fraud detection among financial transactions. In this application the out-of-distribution ‘anomalies’ are the fraudulent transactions that should be identified and flagged. The financial transaction data is processed in a time-series, preserving the auto-correlations between the data. Moreover, the temporal dependence of fraudulent transactions means that unusual patterns develop over time, allowing ML models to capture and exploit their temporal relationship.

There are several challenging aspects to the development of such an AD model for example that, in a similar way to the DQM AD in HEP detectors, anomalies, or cases of fraud, are so infrequent that there is a critical lack of statistics of anomalous events. This rarity and class imbalance can make it difficult to assess the performance of AD models. In the financial industry, less than 1% of credit card transactions are fraudulent, but they can incur huge financial losses on the order of billions of euros [89, 90]. Data in the financial industry, as in many other domains, are protected by strict confidentiality and privacy regulations constraining the size and completeness of datasets. This, alongside the fact that it is very difficult to label data, means that trustworthy labelled data are scarce.

IBM Research France takes a two-fold approach to tackle these challenges, by actively investigating the use of unsupervised models for AD and producing auxiliary synthetic financial transaction datasets. The development of an unsupervised AE-based fraud detection model utilises the modern transformer architecture to encode sequential time series data before a decoder calculates the reconstruction error with respect to the initial input. The AE with iTransformer [91] encoder compares favourably to alternative established classical and NN-based AD methods including MERLIN [92], Isolation Forest [93], fully connected NNs and LSTMs. The performance of transformer-based architectures on numerical time-series data is still an active area of research, particularly the ability of such models to capture long-range dependencies. The interpretation of the reconstruction errors as anomaly scores plays a crucial role in model performance in this setting, various metrics, anomaly thresholds and evaluation protocols (when a fraudulent warning is triggered) are considered.

The production of synthetic financial data is pursued to mitigate the unreliability and lack of labelled data. Unlike HEP, the financial industry does not always have large, open datasets with accurate ground truth labels, it can be necessary for a human expert to manually generate such labels whereas the HEP field relies on advanced simulations underpinned by theoretic predictions of known particle physics phenomena. The application of ML in the financial industry could ameliorate the lack of reliable training statistics, one example is the generation of synthetic data for fraud detection. In a first step, a bank fraud simulation system is built using Markov chains designed to model both legitimate and fraudulent user behaviours within a financial environment. Then, the simulation system leverages large language models in a multi-step generation following nominal and fraudulent strategies with respect of temporal and other dimensions of the described activities captured in the dataset. Whilst this work remains ongoing it demonstrates promising initial results [94].

5.3. Applications in industry: computer vision (CV)

One prominent use of ML models in recent years has been CV; the remarkable efficiency with which NNs can extract and distill high-level information from images and point clouds means that there is a plethora of real-world applications. Many of these applications, like HEP detectors, have strict latency, bandwidth or memory requirements that impose constraints on the deployment of ML models.

The HLS4ML package, as mentioned in section 3, can be used to facilitate the deployment of trained ML models on FPGAs targeting low latency and low power environments. The impact of the project has been felt beyond the wider scientific community and is being used in a variety of different industrial settings, in this section we present two examples.

A joint research project conducted by CERN and Zenseact, explored the use of FPGA hardware accelerators for the real-time semantic segmentation of images from camera sensors on-board autonomous vehicles [95]. The recent advancements in modern deep learning CV applications can improve the safety and efficiency of autonomous driving by allowing the vehicle to analyse and respond to its external environment more accurately and faster.

Using the HLS4ML package, FPGAs were evaluated as a hardware benchmark for CV models based on simple CNN architectures [42]. The ENet architecture [96] was used as a baseline during QAT to attain a suitable compression of the model. An additional constraint limits the total number of filter kernels permitted in the model, similarly to filter ablation, where an optimum value was found by balancing performance and resource requirements.

The inference time of the updated ENet model was tested on a single Zynq Ultrascale+ MPSoC device using a ZCU102 development kit and found to be less than 5 ms per image, faster than previous FPGA implementations. The best performing model was found to use less than 30% of the total resources of the FPGA.

Another example of a real-time CV application is the EU project Edge SpAIce that aims to monitor the plastic pollution in oceans using a satellite-mounted camera and ML techniques. Earth observation satellites are capable of transmitting data back to Earth with a limited bandwidth and power usage. Similarly to the resource constraints faced in HEP detector readout systems, FPGAs can be used to alleviate these issues. Edge SpAIce uses fast NN inference aboard the satellite to segment the images into different object classes such that only images relevant to a particular class are transmitted back to Earth [97]. This classification represents an initial type of bandwidth reduction before data transmission.

As before, the HLS4ML package was used to move the U-net architecture based on CNN layers onto an FPGA hardware accelerator. The task is formulated such that the satellite images ($256 \times 256 \times 3$) are segmented to identify pixels covered by clouds. The supervised training, using per-pixel truth labels, proceeds in a quantisation-aware fashion and uses knowledge distillation from a larger, teacher model to reduce the number of learnable parameters in the smaller, student model. Using the HLS4ML workflow results in a performance improvement of a factor greater than 8 with respect to the Vitis AI tool whilst adhering to the resource constraints of the ZCU102 development kit.

6. Conclusion

The trigger and data acquisition systems at the large LHC experiments have undergone significant changes since LHC Run 1. The capabilities and efficacy of the event selection pipeline have been considerably enhanced by the use of ML techniques. The level of performance achieved in the real-time regime has markedly improved and in many cases approaches the state-of-the-art achieved offline.

Collider experiments and the HEP field more generally have long utilised ML models to ameliorate signal identification. Indeed, with the prodigious datasets provided by MC simulation and real collisions, the LHC has ample training data to exploit and train ML models. However, it is only recently that the trigger systems and real-time reconstruction have been able to leverage these advancements. The challenges of deploying complex algorithms in real-time environments had resisted the progress made for algorithms employed during asynchronous reconstruction.

In spite of the constraints imposed in real-time environments, technological and algorithmic advancements have allowed LHC experiments to deploy ML within their trigger systems. This trend looks set to continue as more applications and opportunities arise. This work presents examples of the usage of ML models in real-time data taking environments across the large LHC experiments. Beyond improving the performance of current algorithms used in the trigger ML in the real-time regime may also extend the capacity of the LHC experiments' physics programmes by offering model-agnostic methods for anomaly detection. Enlarging the phase-space that can be searched by relying on events that exhibit discrepancies with respect to the SM could inform future searches for new physics.

The use of ML in real time is not limited to the HEP field and there are many industrial applications that necessitate similar strategies. The collaboration between CERN and industry partners arising from the unique demands of trigger and data acquisition systems at the LHC have propelled the community forward by providing open, accessible and well supported libraries for hardware deployment of ML. Continuing knowledge exchanges between research and industry will be crucial in order to continue this progress for LHC triggers as well as real-time industrial applications.







Data availability statement

No new data were created or analysed in this study.

Acknowledgments

This work is part of the SMARTHEP network and it is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement N. 956086. We gratefully acknowledge Teng Jian Khoo, Maurizio Pierini and Chris Scheulen for their comments and advice for this whitepaper. We also thank the wider participants of the SMARTHEP network for their insight and feedback.

ORCID iDs

S Astrand  0000-0001-9424-6607
L Bozianu  0000-0002-1243-9980
F I Giasemis  0000-0003-0622-1069
K E Iversen  0000-0001-6533-4085
P Jawahar  0000-0003-2501-249X
H Pineiro Monteagudo  0009-0004-9791-9044

References

- [1] ALICE collaboration 2008 The ALICE experiment at the CERN LHC *JINST* **3** S08
- [2] ALICE collaboration 2014 Performance of the ALICE experiment at the CERN LHC *Int. J. Mod. Phys. A* **29** 1430044
- [3] ALICE collaboration 2019 Real-time data processing in the ALICE high level trigger at the LHC *Comput. Phys. Commun.* **242** 25
- [4] Kvapil J et al 2021 ALICE Central Trigger System for LHC Run 3 *EPJ Web Conf.* **251** 04022
- [5] ATLAS collaboration 2024 The ATLAS experiment at the CERN Large Hadron Collider: A description of the detector configuration for Run 3 *JINST* **19** 05063
- [6] ATLAS collaboration 2008 The ATLAS experiment at the CERN Large Hadron Collider *JINST* **3**
- [7] Hayrapetyan A et al (CMS collaboration) 2023 Development of the CMS detector for the CERN LHC Run 3
- [8] CMS collaboration 2008 The CMS experiment at the CERN LHC *JINST* **3** 1
- [9] CMS Tracker Group collaboration 2020 The CMS Phase-1 pixel detector upgrade (arXiv:2012.14304)
- [10] LHCb collaboration 2008 The LHCb detector at the LHC *JINST* **3** S08005
- [11] LHCb collaboration 2015 LHCb detector performance *Int. J. Mod. Phys. A* **30** 1530022
- [12] Aaij R et al 2022 A comparison of cpu and gpu implementations for the LHCb experiment run 3 trigger *Comput. Softw. Big Sci.* **6** 1
- [13] Aaij R et al 2020 Allen: A high-level trigger on GPUs for LHCb *Computing and Software for big Science* **4** 1
- [14] Grid W L C 2012 The Worldwide LHC Computing Grid (available at: <https://cds.cern.ch/record/1997398>)
- [15] Molina J F, Forti A, Girone M and Sciaba A 2014 Operating the worldwide LHC computing grid: current and future challenges *J. Phys.: Conf. Ser.* **513** 062044
- [16] ALICE collaboration 2014 Upgrade of the ALICE Experiment: Letter Of Intent *J. Phys. G* **41** 087001
- [17] ATLAS collaboration 2024 The ATLAS trigger system for LHC Run 3 and trigger performance in 2022 *JINST* **19** 06029
- [18] CMS collaboration 2020 Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV *JINST* **15** 10017
- [19] CMS collaboration 2017 The CMS trigger system *JINST* **12** 01020
- [20] LHCb collaboration 2024 The LHCb Upgrade I *JINST* **19** 05065
- [21] T.S. Network *Review of triggers in LHC experiments: review of the state of the art of the triggers of LHC collaborations and best practices*
- [22] Smith W H 2016 Triggering at the LHC *Annu. Rev. Nucl. Part. Sci.* **66** 123
- [23] Denby B 1988 Neural networks and cellular automata in experimental high energy physics *Comput. Phys. Commun.* **49** 429
- [24] ALEPH collaboration 1999 Determination of from the measurement of the inclusive charmless semileptonic branching ratio of b hadrons *Eur. Phys. J. C* **6** 555
- [25] OPEL collaboration 1994 A measurement of the production of d mesons on the z0 resonance *Z. Phys. C*
- [26] Lönnblad L, Peterson C and Rönngvaldsson T 1991 Using neural networks to identify jets *Nucl. Phys. B* **349** 675
- [27] Roe B P, Yang H-J, Zhu J, Liu Y, Stancu I and McGregor G 2005 Boosted decision trees as an alternative to artificial neural networks for particle identification *Nucl. Instrum. Methods Phys. Res. A* **543** 577
- [28] Duarte J et al 2018 Fast inference of deep neural networks in FPGAs for particle physics *J. Instrum.* **13** 07027
- [29] Summers S et al 2020 Fast inference of boosted decision trees in FPGAs for particle physics *JINST* **15** 05026
- [30] Umuroglu Y et al 2017 FINN: A framework for fast, scalable binarized neural network inference *Proc. 2017 ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays* pp 65–74
- [31] Hong T, Carlson B, Eubanks B, Racz S, Roche S, Stelzer J and Stumpp D C 2021 Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics *J. Instrum.* **16** 08016
- [32] FastML Team 2024 fastmachinelearning/hls4ml (available at: <https://doi.org/10.5281/zenodo.1201549>)
- [33] Fahim F et al 2021 HLS4ML: An open-source codesign workflow to empower scientific low-power machine learning devices (arXiv:2103.05579)
- [34] Athur D K, Pawar R and Arora A 2025 Out-of-the-box performance of FPGAs for ML workloads using Vitis AI *Int. Symp. on Applied Reconfigurable Computing* (Springer) pp 123–39
- [35] Ahmad J, Jervis M and Venkata R 2022 Inteltextregistered FPGAs and SoCs with Inteltextregistered FPGA AI suite and OpenVINO toolkit drive embedded/edge AI/machine learning applications

- [36] Nagel M, Fournarakis M, Amjad R A, Bondarenko Y, Van Baalen M and Blankevoort T 2021 A white paper on neural network quantization (arXiv:2106.08295)
- [37] Gou J, Yu B, Maybank S J and Tao D 2021 Knowledge distillation: a survey *Int. J. Comput. Vis.* **129** 1789–819
- [38] Cho J H and Hariharan B 2019 On the efficacy of knowledge distillation *Proc. IEEE/CVF Int. Conf. on Computer Vision* pp 4794–802
- [39] Hinton G, Vinyals O and Dean J 2015 Distilling the knowledge in a neural network (arXiv:1503.02531)
- [40] Cheng H, Zhang M and Shi J Q 2024 A survey on deep neural network pruning: Taxonomy, comparison, analysis and recommendations *IEEE Trans. Pattern Anal. Mach. Intell.* **46** 10558–78
- [41] Kuzmin A, Nagel M, Van Baalen M, Behboodi A and Blankevoort T 2023 Pruning vs quantization: Which is better? *Advances in Neural Information Processing Systems* vol 36 p 62414
- [42] Aarrestad T et al 2021 Fast convolutional neural networks on FPGAs with HLS4ML *Mach. Learn.: Sci. Technol.* **2** 045015
- [43] Coelho C N, Kuusela A, Li S, Zhuang H, Ngadiuba J, Aarrestad T K, Loncar V, Pierini M, Pol A A and Summers S 2021 Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors *Nat. Mach. Intell.* **3** 675
- [44] Wang E, Davis J J, Moro D, Zielinski P, Lim J J, Coelho C, Chatterjee S, Cheung P Y and Constantinides G A 2021 Enabling binary neural network training on the edge *Proc. 5th International Workshop on Embedded and Mobile Deep Learning* pp 37–38
- [45] Pappalardo A X 2023 (available at: <https://doi.org/10.5281/zenodo.3333552>)
- [46] ONNX Runtime developers 2018 ONNX runtime
- [47] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R B, Guadarrama S and Darrell Y 2014 Caffe: Convolutional architecture for fast feature embedding (arXiv:1408.5093)
- [48] Paszke A 2019 PyTorch: An imperative style, high-performance deep learning library (arXiv:1912.01703)
- [49] Pedregosa F et al 2011 Scikit-learn: Machine learning in Python *J. Mach. Learn. Res.* **12** 2825
- [50] M A et al 2015 TensorFlow: Large-scale machine learning on heterogeneous systems
- [51] Hoecker A et al 2007 TMVA-toolkit for multivariate data analysis (arXiv:physics/0703039)
- [52] Sculley D et al 2015 Hidden technical debt in machine learning systems *Advances in Neural Information Processing Systems* vol 28
- [53] Cohen T and Welling M 2016 Group equivariant convolutional networks *Int. Conf. on Machine Learning* (PMLR) pp 2990–9
- [54] Spinner J, Bresó V, De Haan P, Plehn T, Thaler J and Brehmer J 2024 Lorentz-equivariant geometric algebra transformers for high-energy physics *Advances in Neural Information Processing Systems* vol 37 p 22178
- [55] Toscano J D, Oommen V, Varghese A J, Zou Z, Daryakenari N A, Wu C and Karniadakis G E 2025 From pinns to pikans: Recent advances in physics-informed machine learning *Mach. Learn. Comput. Sci. Eng.* **1** 1
- [56] Shlomi J, Battaglia P and Vlimant J-R 2020 Graph neural networks in particle physics *Mach. Learn.: Sci. Technol.* **2** 021001
- [57] Wetzel S J, Ha S, Iten R, Klopotek M and Liu Z 2025 Interpretable machine learning in physics: A review (arXiv:2503.23616)
- [58] Arzani A, Yuan L, Newell P and Wang B 2024 Interpreting and generalizing deep learning in physics-based problems with functional linear models *Eng. Comput.* **1**
- [59] Guest D, Cranmer K and Whiteson D 2018 Deep learning and its application to LHC physics *Annu. Rev. Nucl. Part. Sci.* **68** 161
- [60] Morris A 2024 First experiences with the LHCb heterogeneous software trigger (arXiv:2412.05041)
- [61] Correia A, Giasemis F, Garroun N, Gligorov V V and Granado B 2024 Graph neural network-based track finding in the LHCb vertex detector *JINST* **19** 12022
- [62] Giasemis F I, Lončar V, Granado B and Gligorov V V 2025 Comparative analysis of FPGA and GPU performance for machine learning-based track reconstruction at LHCb (arXiv:2502.02304)
- [63] ATLAS collaboration 2024 The ATLAS trigger system for LHC Run 3 and trigger performance in 2022 (arXiv:2401.06630)
- [64] ATLAS collaboration 2020 An ensemble of neural networks for online electron filtering at the ATLAS experiment *J. Phys.: Conf. Ser.* **1525** 012076
- [65] CMS collaboration 2023 CMS High Level Trigger Performance for Run 3 *Technical Report* 449 (CERN)
- [66] CMS collaboration 2022 Identification of hadronic tau lepton decays using a deep neural network (arXiv:2201.08458)
- [67] ATLAS collaboration 2023 Fast b-tagging at the high-level trigger of the ATLAS experiment in LHC Run 3 *JINST* **18**
- [68] ATLAS collaboration 2025 Configuration, performance and commissioning of the ATLAS b-jet triggers for the 2022 and 2023 LHC data-taking periods *JINST* **20**
- [69] ATLAS collaboration Transforming jet flavour tagging at ATLAS (arXiv:2505.19689)
- [70] Collective 2024 The Large Hadron Collider and The Experiments for Run 3 *JINST* **19**
- [71] Komiske P T, Metodiev E M, Nachman B and Schwartz M D 2017 Pileup mitigation with machine learning (PUMML) *J. High Energy Phys.* **2017** 1
- [72] Maier B, Narayanan S M, de Castro G, Goncharov M, Paus C and Schott M 2022 Pile-up mitigation using attention *Mach. Learn.: Sci. Technol.* **3** 025012
- [73] Delaney B, Schulte N, Ciezarek G, Nolte N, Williams M and Albrecht J 2024 Applications of lipschitz neural networks to the Run 3 LHCb trigger system *EPJ Web Conf.* **295** 09005
- [74] Schulte N, Delaney B R, Nolte N, Ciezarek G M, Albrecht J and Williams M 2023 Development of the topological trigger for LHCb Run 3 (arXiv:2306.09873)
- [75] Allen N et al 1998 Measurement of R_b using a vertex mass tag *Phys. Rev. Lett.* **80** 660
- [76] ALICE collaboration 2021 Reconstruction in ALICE and calibration of TPC space-charge distortions in Run 3 (arXiv:2109.12000)
- [77] Gorbunov S et al 2021 Deep neural network techniques in the calibration of space-charge distortion fluctuations for the ALICE TPC *EPJ Web Conf.* **251** 03020
- [78] Ronneberger O, Fischer P and Brox T 2015 U-net: Convolutional networks for biomedical image segmentation *Miccai* (Springer) pp 234–41
- [79] Belis V, Odagiu P and Aarrestad T K 2024 Machine learning for anomaly detection in particle physics *Rev. Phys.* **12** 100091
- [80] Pol A A, Cerminara G, Germain C, Pierini M and Seth A 2019 Detector monitoring with artificial neural networks at the cms experiment at the cern large hadron collider *Comput. Softw. Big Sci.* **3** 1
- [81] Brinkerhoff A et al 2025 Anomaly detection for automated data quality monitoring in the cms detector (arXiv:2501.13789)
- [82] CMS ECAL collaboration 2024 Autoencoder-based anomaly detection system for online data quality monitoring of the CMS electromagnetic calorimeter *Comput. Softw. Big Sci.* **8** 11
- [83] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 770–8

- [84] ATLAS collaboration 2024 Autoencoder-based time series anomaly detection for ATLAS Liquid Argon calorimeter data quality monitoring *Technical Report* 002 (CERN)
- [85] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
- [86] CMS collaboration 2024 Real-time anomaly detection at the L1 trigger of CMS experiment (arXiv:2411.19506)
- [87] CMS collaboration 2023 Level-1 Trigger Calorimeter Image Convolutional Anomaly Detection Algorithm CMS-D-2023-086 Tech. Rep. CERN
- [88] CMS collaboration Anomaly Detection in the CMS Global Trigger Test Crate for Run 3
- [89] Borgne Y-A L, Siblini W, Lebichot B and Bontempi G 2022 *Reproducible Machine Learning for Credit Card Fraud Detection - Practical Handbook* (Université Libre de Bruxelles)
- [90] Bank E C *Sixth Report on Card Fraud*
- [91] Liu Y et al 2023 itransformer: Inverted transformers are effective for time series forecasting (arXiv:2310.06625)
- [92] Nakamura T, Imamura M, Mercer R and Keogh E 2020 Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives 2020 *IEEE Int. Conf. on Data Mining (ICDM)* (IEEE) pp 1190–5
- [93] Liu F T, Ting K M and Zhou Z-H 2008 Isolation forest 2008 *8th IEEE Int. Conf. on Data Mining* (IEEE) pp 413–22
- [94] Feillet P and Olocco M 2025 *GeNovae*
- [95] Ghielmetti N et al 2022 Real-time semantic segmentation on FPGAs for autonomous vehicles with HLS4ML *Mach. Learn.: Sci. Technol.* **3** 045011
- [96] Paszke A, Chaurasia A, Kim S and Culurciello E 2016 Enet: A deep neural network architecture for real-time semantic segmentation (arXiv:1606.02147)
- [97] Tzelepis S, Ghielmetti N, Lemoine N-M, Pierini M, Summers S and De Vielleville F 2024 Edge SpAIce: enabling onboard data compression with machine learning on FPGAs (available at: <https://doi.org/10.5281/zenodo.13865939>)