ISSN: 1813-4890

Design and Implementation of Intelligent Detection for Abnormal User Storage Access Behavior in High Energy Physics Computing Environments

Yuhan Mu

Oregon Episcopal School, Portland, OR 97223, USA.

cathymumm@163.com

Abstract

High energy physics (HEP) computing clusters run tremendous amounts of concurrent jobs across a distributed, multi-layer storage. In this multi-user, multi-job environment, access patterns to storage become highly dynamic, and static thresholds and rule-based monitoring fail to discover anomalies; a few abnormal accesses can disrupt normal access behavior, causing degraded performance or cascading failures. This paper describes an integrated monitoring and intelligent detection architecture for abnormal user storage access behavior. Lightweight probes are deployed on computing, storage, and scheduling nodes that collect input/output (I/O) throughput and input/output operations per second (IOPS), as well as job and user identifiers, followed by cleaning, aggregation, and indexing in an Elasticsearch, Logstash, and Kibana (ELK) Stack analytics platform that allows fast query and visualization. For detection, an unsupervised Isolation Forest model is chosen both to avoid reliance on scarce numbers of labeled anomalies and to cope with high-dimensional features. Production validation under tens of thousands of concurrent jobs shows rapid detection and localization to specific jobs and users, allowing diagnosis and resolution much faster than previously possible and improving overall platform stability.

Keywords

High energy physics computing, storage anomaly detection, ELK Stack, Isolation Forest, storage behavior monitoring.

1. Introduction

1.1. Research Background and Scientific Significance

High-Energy Physics (HEP), or particle physics, is one of the primary directions of research into understanding the basic physics of the universe. HEP experiments have been conducted using large, complex particle colliders and supporting equipment, and they produce experimental data that are critical for validating the Standard Model and for the investigation of new physics. HEP experiments tend to have long construction periods, long operational lifespan, and extensive collaboration internationally between many research organizations, institutes and scientists. The data produced by HEP experiments represent a very important component of the branch of fundamental physics.

Several high-energy physics and astroparticle physics experiments of international scope have been launched and are being built in China and around the world. The Beijing Spectrometer III (BESIII) Experiment aims at precision measurement in light hadron spectroscopy, charm physics, and τ lepton properties, as well as testing quantum chromodynamics and studying the low energy regime of strong interactions [1]. The Jiangmen Underground Neutrino Observatory (JUNO) is designed to carry out ultrahigh precision reactor antineutrino energy spectra

ISSN: 1813-4890

measurements with the ultimate goal of determining the neutrino mass ordering as well as accurately measuring the neutrino oscillation parameters. It is one of the most important neutrino experiments in the world at present [2]. Other examples include the long-term monitoring of ultra-high-energy gamma rays and cosmic rays by the Large High Altitude Air Shower Observatory (LHAASO) using cosmic ray showers detected by a distributed array of ground stations with the aim of understanding their origin and what acts as their engine, as well as providing new capabilities for researchers working in the area of high energy astrophysics [3]. While these experiments may differ in their science case, they all have extremely demanding requirements in respect of large scale data processing and computational analysis.

With the increasingly high precision of the detector and the performance of the trigger system, the amount of data output is likewise growing rapidly from year to year; the data scale of a single experiment ranges from the terabyte (TB) level in early stages to the petabyte (PB) or even exabyte (EB) level today; several TB of data must be processed and analysed. In this context, it is clear that computing plays an increasingly central role in computational procedures, shifting from computer 'auxiliary' to 'main character', so that the experimental research paradigm changes from 'experiment-driven' to 'computation-driven'.

Hence, compute and storage systems and their methodology-based software infrastructure are now regarded as a "third class of experimental facilities" on par with accelerators and detectors, with the efficiency and reliability of their operation directly impacting the quantity of data produced by experiments and the resulting science.

1.2. Challenges in Computing Clusters and Storage Systems

Modern high energy physics computing systems are typically built up from distributed computing nodes collaborating with multi-tier storage systems to provide the desired mix of compute and access performance for a disparate set of workloads that differ in their requirements for computational power and data access performance. A typical storage setup might consist of a number of layers of storage types ranging from local disks to fast distributed file systems and finally to long-term storage to serve a vast population of users and jobs. Although these architectures yield significant overall gains in computing efficiency, they introduce significant complexity into the system runtime environment as well.

In multi-user, multi-job concurrent scenarios, storage access patterns are very dynamic and chaotic, and its input/output (I/O) characteristics can vary greatly among different users and different analysis tasks. Under such conditions, some existing storage monitoring techniques based on static thresholds or rules may fail to detect abnormal behaviours in time, leading to one or two individual jobs or users triggering performance degradation of the storage system and even cascading faults to the execution of other correct normal jobs, endangering the operation of the whole computing service.

Thus, conducting fine-grained monitoring of the storage systems themselves and enabling fast localization and accurate identification of abnormal behaviors occurring under complex computing architectures become the pain points of operation and management for the high energy physics computing platforms.

1.3. Research Problems and Contributions

Inspired by the aforementioned challenges, this paper studies the problem of storage anomaly detection in computing clusters used in high energy physics, focusing on automated and near real-time anomaly identification in real production settings. The research questions essentially comprise: How can enormous volumes of operational data in a large-scale distributed system be efficiently collected and managed by the system? How can machine learning techniques be

ISSN: 1813-4890

effectively applied to latent abnormal patterns discovery from complex high-dimensional data to ensure stable operation of computing services?

The main contributions of this work can be summarized as follows. First, this work proposes an integrated storage monitoring and anomaly detection architecture for high energy physics computing clusters enabling unified collection and analysis of storage system operational states. Second, an extensible big data processing platform based on the Elasticsearch, Logstash, and Kibana (ELK) Stack is designed and implemented to facilitate concurrent storing, indexing, and querying of large amounts of data. Third, this work presents an unsupervised anomaly detection approach based on the Isolation Forest algorithm that detects abnormal storage behaviours without relying on manually labelled training data. Finally, the method is validated in a real large-scale concurrent job execution environment, and the results show that the solution effectively identifies abnormal jobs and user behaviours, supporting system stability and operational efficiency.

2. System Architecture

The predominant style of large scale computing system for high energy physics experiments is that of a large distributed computing cluster. It is a responsibility of this cluster to provide the core facility's data processing and analysis application as well as simulations. The system architecture is described from different perspectives, including user access, job scheduling, and data storage, as well as user behavior. It is worth analyzing what the main factors are that destabilize the system, as this provides an idea of the base architecture over which the storage anomaly discovery techniques are described.

2.1. User-Oriented Computing Service Architecture

The principle "users are not interested in how it works" applies to the high energy physics use case: the primary people using the computing "platform" are experimental physicists and data analysts. These people are not too concerned about the details of how the system works, but rather focused on how to submit jobs, how quickly those jobs execute, and the reliability of the analysis that they produce. As a result, the "club" or computing system must provide a common, simple, and efficient means of access.

In real operation scenarios, users usually connect to the computing service via a Web Portal or a Command Line Interface (CLI). The Web Portal is aimed more at novice users or interactive use cases and provides visualized job management as well as status monitoring. The CLI is more appropriate for batch job submission and automated analysis pipelines and is widely used for "Big Data" analyses. Secure shared access to computing and storage is facilitated by unified authentication and authorization. The workflow for job submission is described in Figure 1.

This stage of the life cycle is often referred to as the job submission process, whose actual steps vary from scheduling system to scheduling system. They typically consist of job description, resource request, and finally task dispatching. When users submit a job for execution, they must describe what resources are being requested; the number of central processing unit (CPU) cores, the size of memory, and so on, and they must also give the locations of input and output data sets and the scripts that read and write them. Upon receiving the job request, the scheduler must parse the job description, and it must dispatch the job to the appropriate computing nodes, depending on the current configuration of the cluster computers.

ISSN: 1813-4890

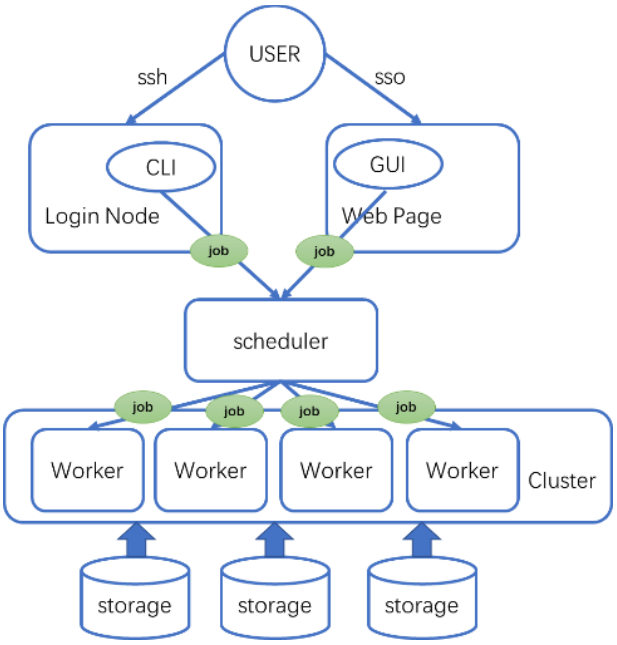


Figure 1: Workflow of high-energy physics computing job submission.

2.2. Job Scheduling and Dispatching Mechanism

The job scheduling system is the totality of scheduling components. The scheduler itself is usually resource-aware, able to get real-time information about resource use throughout the compute nodes, and dispatch new jobs in a timely fashion when there are idle job slots available. If users and tasks cannot be treated equally, the scheduling system may also implement multiple queues and priority levels.

A job generally goes through phases of submission, queuing, execution and completion or failure. In high-throughput computing, the scheduler is faced with scheduling thousands of jobs at once in the various invocations of these phases, placing great pressure on scheduling capacity and system reliability of the machines serving jobs. In high energy physics experiments in particular, there may be tens of thousands of jobs running at any one time and producing bursty I/O requests at startup, during execution, and at termination.

A co-occurrence of highly-concurrent and dense jobs leads to stress on the storage system, since the impact of any small aberration in one job's I/O behavior quickly propagates to other jobs and degrades system performance. Keeping track of the storage access behavior of jobs in execution is a non-trivial requirement for stable operation.

2.3. Distributed Storage System for Experimental Data

In order to satisfy the massive data storage and access requirements of high energy physics experiments, computing systems generally adopt a hierarchical distributed storage architecture. Distributed file systems are used to store the experimental data requiring long-term preservation, such as the job input data and the final output results. Such a system can give high reliability and high scalability. A local cache storage deployed at the computing nodes is used to store the intermediate files produced while executing jobs. Such data generally need not be kept very long, but require high I/O performance.

In practice, the storage system has to accommodate many concurrent access patterns at all times, such as sequential read and write, random read and write as well as frequent metadata operations. Among these include continuous or batch reads and writes of raw data, creating or deleting files and folders, probing or setting attributes on files, traversing directories and checking access permissions. The sequential access pattern is more commonly found in the workloads that involve massive data processing. Conversely, the random access and metadata

reads and writes are more related to analysis workloads and job scheduling. The interleaving of the various access patterns results in complex and dynamically shifting load characteristics.

2.4. Analysis of User-Induced Instability Factors

Compared with hardware and software aspects, user behaviour is an additional source of instability. In multi-user shared environments, there is a large degree of variation from one user to another, both in respect of job structure, and analysis techniques. Some jobs will exhibit unusual patterns of storage access, due to bad programme parameters, or defects in the programme logic itself; unexpected large concurrent I/O operations, excessive storage access to small files, and abnormal random read and write operations will provide further stress.

Because such abnormal behaviours are sometimes not symptoms of system failures but originate from user-level actions, conventional fault detection techniques have great difficulty in quickly pinpointing the cause, for instance, the “guilty” user. If the abnormal jobs are undetected and unaddressed, then there may be leakage in storage performance, which will also affect the execution of normal users’ jobs.

2.5. Stability Assurance Requirements

Based on the above analysis, high energy physics computing systems impose increasingly stringent requirements on stability assurance. First, real-time monitoring of computing and storage resources is required to maintain timely awareness of system operating conditions. Second, the system should be capable of rapidly locating anomaly sources and accurately associating abnormal behaviors with specific jobs or users. Finally, automated analysis techniques are needed to reduce reliance on manual operations and improve overall operational efficiency. These requirements provide a clear application background for the introduction of big data analytics and machine learning–based anomaly detection methods in subsequent sections.

3. Design of Anomaly Detection Methods

To address the challenges posed by the complex operating environment of storage systems, large-scale job concurrency, and the difficulty of timely anomaly identification in high energy physics computing clusters, this paper proposes a storage anomaly detection solution based on a big data analytics platform and machine learning algorithms. This chapter focuses on the overall design philosophy of the proposed approach and the detailed implementation of the big data analytics architecture.

3.1. Overall Design Philosophy

In high-throughput computing environments, a storage anomaly detection system is required to continuously collect and analyze large volumes of operational data without adversely affecting existing computing services. Therefore, during the design phase, clear objectives are established in terms of system scalability, intrusiveness, and analysis timeliness.

First, with respect to scalability, the system must be capable of accommodating a gradual expansion of job concurrency from the current level of approximately 40,000–50,000 concurrent jobs to a future scale of millions of jobs. This requires that all components — data collection, data transmission, data storage, and data analysis — exhibit a strong degree of horizontal scalability and actively avoid introducing single points of failure or bottlenecked performance paths. Next, with respect to intrusiveness, it is required that the monitoring and analysis mechanisms impose minimal overhead on the existing computing and storage services; the data collection process should be light enough that interference with normal business applications does not occur. Finally, with regard to the timeliness of analysis, it is desirable that the system supports near real-time transfer and detection of anomalies, completing the tasks

ISSN: 1813-4890

of data aggregation and analysis in a window of approximately 10–20 minutes so that the detected anomaly can be used for decision support for both the system operators and the scheduling components.

For the above design goal, this paper considers the technical approach of a big data platform with machine learning–based anomaly detection. By building a unified data collection and analysis platform, the operational data in a distributed computing environment are centralized; in addition, machine learning is leveraged in order to automatically detect anomalies under complex backgrounds and hence boost overall system stability and operational efficiency.

3.2. Big Data Analytics Architecture Design

In very large-scale distributed systems operational data are usually voluminous, heterogeneous and structured by complex schemas. To satisfy the requirements of high-concurrency data ingestion, flexible querying, and visual analytics, the ELK Stack is adopted as the core technology framework of the big data analytics platform [4].

3.2.1. Data Collection Layer

The data collection layer is the base of the entire analytics system. Its primary purpose is to gather at runtime metrics related to storage access behaviour, without measurable disruption to the normal business of the system under study. Given the architectural features of high energy physics computing systems, lightweight probe programs are used, and multiple probe programs are deployed at a number of salient points so that the collection process has coverage across several layers of the system.

Probes are deployed most frequently at three classes of nodes. At computing nodes, probes collect I/O behavior generated by job execution and associate that behavior with jobs using triggering processes. At storage nodes, probes observe storage I/O characteristics and triggering events related to those characteristics. At job scheduling nodes, probes get user information tied to triggering processes, and through judicious deployment of the probes, low-level storage access behavior can be tied to specific users and job instances.

In terms of metrics selection, this work emphasizes key metrics that can reflect load and access patterns on the storage (for example, storage I/O throughput and input/output operations per second (IOPS), together with job- and user-level identifiers). The former metrics help characterize the load that jobs put on the storage system, while the latter metrics provide meaningful contextual information for naming the anomalous. All data collected above are wrapped up in a structured format and sent through a single unified pipeline to the backend analytics platform.

3.2.2. Data Preprocessing

Since the raw data are collected from varied nodes and heterogeneous sources, the sources may differ in completeness and data quality levels. Hence, systematic preprocessing is needed before the data reach the analysis phase. Data preprocessing mainly consists of data cleaning and data aggregation.

During the data cleaning process, obvious outliers are first discarded, for example, outliers caused by probe initialization, abnormal stops, and so on. Later, missing data are dealt with by interpolation or labeling. This step is taken to ‘fill up’ the time series data, so that later analytical phases would not be affected.

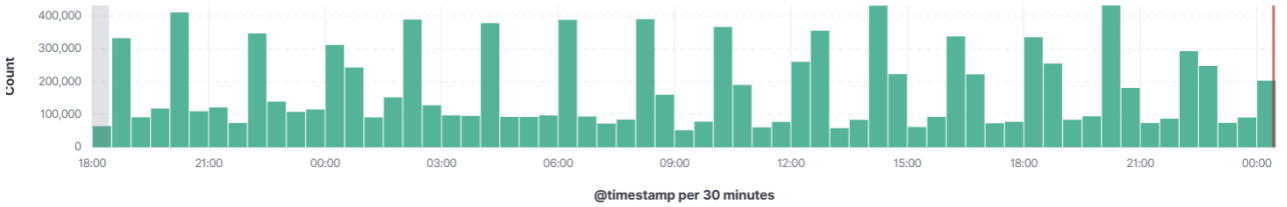
In the data aggregation stage, statistical summarization of collected data according to selected time windows takes place producing feature representations at various temporal scales. In the meantime, data are aggregated along the user and job dimensions in order to model the storage access behavior of different users and jobs. This multi-dimensional aggregation framework allows not only for reduced data volume but also serves to foreground abnormal behaviors against a background of the normal.

ISSN: 1813-4890

3.2.3. Indexing and Storage Design

After preprocessing, the data are written into an Elasticsearch cluster for indexing and storage. Index design is crucial for query performance and system scalability, and this work focuses on the design of the index along three dimensions: time, job and user identifiers, and sets of storage access behaviour metrics.

Time-based indices are designed to support fast queries over specific time ranges, satisfying the requirements of near real-time analysis as well as historical data backtracking. User-oriented indices enable the system to rapidly locate and analyze behavior patterns associated with specific users, while job-oriented indices support fine-grained analysis of individual job execution processes. Through the combined design of multi-dimensional indices, the system is able to maintain good query performance and analytical efficiency even under high-concurrency data ingestion. The index structure is shown in Figure 2. This provides a stable and reliable data foundation for subsequent machine learning–based anomaly detection.



Time	fsname	operation	trigger	value
Feb 6, 2026 @ 00:25:29.000	cefs	sync	python3_24095	0.05
Feb 6, 2026 @ 00:25:29.000	cefs	setattr	sad1_exe_23451	0.183
Feb 6, 2026 @ 00:25:29.000	cefs	close	libuv-worker_24154	0.95
Feb 6, 2026 @ 00:25:29.000	cefs	getattr	libuv-worker_22703	0.017
Feb 6, 2026 @ 00:25:29.000	cefs	getattr	sh_21314	0.017
Feb 6, 2026 @ 00:25:29.000	cefs	close	sub_eeHH_4b_m60_24054	0.017
Feb 6, 2026 @ 00:25:29.000	cefs	getattr	condor_starter_23924	0.233

Figure 2: Key index fields used to store user storage access behavior information.

3.3. Machine Learning–Based Anomaly Detection Method

In the operational environment of high energy physics computing clusters, abnormal storage access behaviors are typically characterized by strong burstiness, short duration, and scarcity of abnormal samples. Moreover, comprehensive manual labeling of such anomalies is generally infeasible in practice. Therefore, adopting machine learning techniques to automatically detect abnormal storage access behaviors has become an important technical approach for improving system stability. This section first reviews commonly used anomaly detection algorithms, then analyzes algorithm selection based on the characteristics of high energy physics computing scenarios, and finally presents the detailed design of an Isolation Forest–based anomaly detection model.

3.3.1. Overview of Representative Machine Learning Algorithms

For anomaly detection tasks, existing machine learning approaches can generally be categorized into statistical-based methods, distance- or clustering-based methods, and model-based methods. This work focuses on several representative algorithms that are commonly used in engineering practice.

(1) K-Means Clustering Algorithm

K-Means is a classical unsupervised clustering algorithm [5] that clusters the data into multiple clusters through minimisation of distance between samples and cluster centroid; in the application of anomaly detection, samples far from their appropriate cluster centre are then

detected as novelties. The algorithm is simple to implement and relatively fast to compute, allowing it to be run on large volumes of data. The main limitation of K-Means clustering is that it requires the number of clusters to be provided a priori, it is also responsive to initialisation of cluster centroids, and, for complex distribution of data, can be poor at modelling non-spherical distributions. In the data structure problem of high energy physics computing, storage access behaviours are often highly non-linear and heterogeneous, limiting the applicability of K-Means to anomaly detection.

(2) Support Vector Machine (SVM)

Support Vector Machines, particularly One-Class SVMs, are popular approaches to unsupervised anomaly detection [6]. This method draws an enclosing decision boundary in the high-dimensional feature space around most of the normal samples and considers outliers as anomalies. One-Class SVMs have certain advantages for operating in the small-sample high-dimensional realm. However, the training process is notoriously sensitive to parameter selection and quickly becomes computationally prohibitive for large datasets, rendering them perhaps shallow in difficult high-throughput computing environments.

(3) Random Forest

Random Forest is another ensemble algorithm that can be used in a supervised or semi-supervised fashion by creating multiple decision trees and combining their outputs by voting or weighted averaging [7]. When used for anomaly detection, the Random Forest model will usually rely on the availability of some labeled abnormal samples to perform well. It can still learn complex relationships in the features. In this scenario (high energy physics computing clusters), abnormal samples are available only on a limited scale (from a volume point of view) and differ in appearance, while high quality labeled data are also scarce.

(4) Isolation Forest

Isolation Forest is an unsupervised algorithm for anomaly detection. The basic idea is that one of the most efficient ways to isolate an anomaly is by randomly selecting a feature and a split [8]. When compared to distance- or density-based methods, it does not need to explicitly model what the “normal” looks like, capitalising instead on the fact that abnormal samples are easier to isolate. It is computationally cheap and robust against high-dimensional and large-scale datasets, making it worth considering in production.

3.3.2. Algorithm Selection Analysis

Storage anomaly detection for physics jobs on HEP computing clusters is interesting for a number of reasons. First, in normal operation the majority of jobs exhibit expected storage access patterns and only a very small fraction of jobs contain abnormal behaviour. Second, abnormal samples are diverse in nature and not all conform to the same labelling standard. This implies that anomaly detection approaches should not rely heavily on labelled aberrant samples and should generalise strongly.

From the viewpoint of algorithm applicability, clustering-based methods such as K-Means come with low computational cost, but have low expressivity with respect to complex data distributions. One-Class SVMs provide good theoretical anomaly detection properties, but their training and parameter tuning costs scale highly with respect to large size, and in a high concurrency context, it may be hard to convince users to seek out labels for them. More generally, supervised learning methods such as Random Forest depend on an abundance of labelled data, whereas sufficient labels are not available for the domain targeted.

In a thorough assessment of the pros and cons of these methods, Isolation Forest was selected as the core model to experiment with for storage. The reasons are manifold, but chiefly:

It is a completely unsupervised model not requiring labelled abnormal samples, which is useful in settings where the actual number of abnormalities is quite small. It leverages higher-dimensional learning well. The storage access records are made up of a multitude of

ISSN: 1813-4890

performance metrics, so this is an important consideration. It does not have vast computational complexity and can be implemented in parallel. This is important so that it can be accessed in near real time on a high energy physics cluster.

3.3.3. Design of the Isolation Forest–Based Anomaly Detection Model

In the specific model design, this study focuses on three key aspects: feature selection, model training, and anomaly scoring with threshold determination.

(1) Feature Selection

Feature selection is another essential consideration during anomaly detection model design, the rationality of which directly affects the model's sensitivity to abnormal behaviours and its general ability to generalize. In the storage monitoring scenario of high-energy physics (HEP) computing clusters, anomalous behaviour usually manifests as a relatively large deviation between the storage access behaviour of a job and the characteristics of its normal workload. Accordingly, this study emphasizes the extraction of the indicators from the storage access behaviour that best reflect the job execution state and workload features.

Specifically, this study chooses I/O throughput and IOPS in a unit time interval as the basic features to describe the bandwidth demand and operation frequency of jobs on storage systems. Furthermore, to help the model glean a better semantic understanding of abnormal behaviors, contextual information about jobs and users, such as job identifiers and user identifiers, are also incorporated to capture differences in access behavior for different jobs and users. Thus, the storage access patterns of jobs being executed are characterised using storage performance metrics augmented with contextual information.

In these procedures for feature construction, raw metrics are first normalized to avoid different units and different ranges from skewing the training of the model. Then multiple indicators are combined into a multidimensional feature vector, thereby allowing the representation of the characteristics of the storage behaviour of different jobs in the feature space. Such a multidimensional representation of features allows brushing up the ability of model to detect complex patterns of anomalies.

(2) Model Training Process

Isolation Forest training: takes historic runtimes as input then learns storage access behaviors under the assumption that normal ain't in the minority. There's no manually labelled data - the training phase learns the gist of normal distribution (the yay way) through randomisation, something useful in high energy physics computing which is built on the idea of strange things happening.

During model training, the Isolation Forest randomly selects a fixed number of feature dimensions, as well as a random split threshold, to create a number of independent "isolation trees" layer by layer. In each isolation tree, the samples are level-wise randomly partitioned recursively onto branch nodes, until all samples are isolated, or until some pre-defined maximum depth is reached. As the anomalous samples are usually "scarce" in the feature space, they are typically contained by fewer random splits than normal sample points.

During inference work, the average path length is calculated from each sample to quantify the "anomaly degree". The whole training process is rather low-complexity, and naturally allows parallelization for rapid deployment in big data consumption scenarios. The model can also be periodically re-trained on the latest runtime data each time a sliding time window moves, helping it to track the drift in cluster operating conditions and workload patterns in a dynamic manner.

(3) Anomaly Scoring and Threshold Determination

At inference time, the Isolation Forest assigns an anomaly score to each sample input, indicating how far the sample lies from the distribution of normal behaviour in the feature space. Broadly,

ISSN: 1813-4890

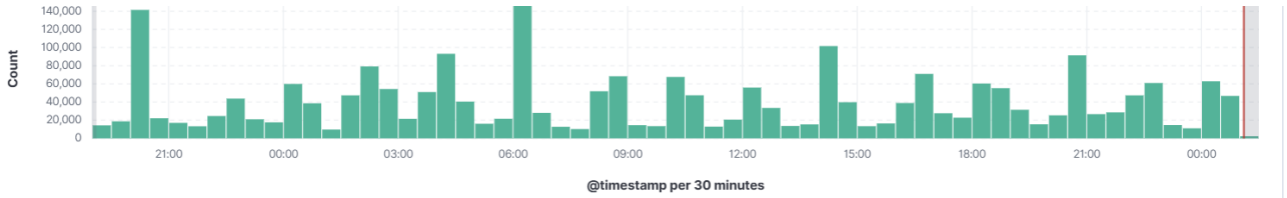
the more anomalous a sample is, the shorter its average path length, and thus, higher its anomaly score. Normal samples tend to have lower scores.

To enable both stable and reliable anomaly identification in practice, an adaptive threshold-setting strategy is designed based on the statistical distribution of the identified anomaly scores. In particular, anomaly scores from a given time period are summarised to determine a requirement-dependent dynamic threshold range that mitigates misclassification due to fluctuations in workloads and operational modes.

If the anomaly score from the job, or from the particular time window, exceeds the threshold then that job is marked as a potential anomaly, specifically related to a job and a user. Such an anomaly detection result therefore not only indicates the presence of any abnormal behaviour, but provides evidence for anomaly localisation, user behaviour analysis and for the purpose of general operational decision.

4. Anomaly Detection Effectiveness Analysis

Based on the anomaly detection workflow described above, the system performs feature construction and modeling analysis on the collected storage access behavior data. Through unsupervised learning and clustering analysis of multi-dimensional features, the model is able to characterize the normal storage access patterns of different jobs and users during regular operation, and further quantify the degree to which their behaviors deviate from the normal distribution. As a result, anomaly scores representing user behavior abnormalities are generated, as illustrated in the Figure 3.



Time	fsname	trigger	lastday_score	lastweek_score	username
Feb 7, 2026 @ 01:00:01.000	acfs	run_submit_sh_24015	0.305	0.298	rfzhang
Feb 7, 2026 @ 01:00:01.000	acfs	DelphesHepMC2_24015	0.306	0.298	rfzhang
Feb 7, 2026 @ 01:00:01.000	acfs	starter-suid_10532	0.312	0.306	zhangyt
Feb 7, 2026 @ 01:00:01.000	acfs	python3_21880	0.338	0.3	wanghan
Feb 7, 2026 @ 01:00:01.000	acfs	run_sh_24015	0.305	0.298	rfzhang
Feb 7, 2026 @ 01:00:01.000	acfs	starter-suid_10957	0.305	0.304	xial

Figure 3: User behavior anomaly scores generated by the proposed method.

When the cluster operates under high load and executes a large number of concurrent jobs, once a sudden degradation or abnormal fluctuation in storage performance occurs, the system can automatically rank the currently running jobs and users based on their anomaly scores. Higher anomaly scores indicate a greater deviation of storage access behavior from normal patterns and, consequently, a higher potential anomaly risk. By analyzing the ranked anomaly scores, system administrators can efficiently identify key users or job instances that are most likely to be responsible for storage performance issues among a large number of running tasks.

By associating the anomaly scores with contextual information such as job and user identifiers (IDs) and time windows of access, the system not only allows for detecting abnormal storage access behavior (that is, efficiently pinpointing the abnormal behavior while shortening the time of manual troubleshooting) but also supports analysis of the duration of such access and the scope of how many storage accesses this misbehavior impacts and influences cluster performance. The score is transformed into a usage context based on these IDs and time

windows, which can benefit the stability and accurate monitoring of large-scale computing clusters.

5. Summary of the Work

With the increasing amount of data and computer resources consumed by high-energy physics experiments, computer and storage stability have long been prerequisites for the successful execution of experiments. Guidelines that operators relied on to operate and maintain HEP computing and storage clusters have been shown to have deficiencies in that they respond to real problems too late, or don't cover enough problems through the use of static memory and experience. To tackle this practical problem, this paper implements and proposes an automated storage anomaly detection system based on big data analytics and machine learning designed for high-energy physics computing clusters.

This work first analyzes the complexity and hidden nature of storage anomalies under distributed computing and multi-tier storage architectures from the perspective of large-scale high-energy physics experiments. On this basis, a unified monitoring architecture covering compute nodes, storage nodes, and scheduling nodes is constructed to enable multidimensional data collection and correlation analysis of storage access behaviors. By introducing a lightweight probing mechanism, fine-grained characterization of job-level and user-level storage behaviors is achieved with minimal system intrusion, providing a reliable data foundation for anomaly detection.

At the data processing and analysis level, a scalable big data analytics platform based on the ELK Stack is designed and implemented, covering the whole flow from data collection and data preprocessing, to indexed storage, querying and visualization, maintaining stable behavior during concurrent insertions of data, and allowing flexible queries by various dimensions - time, jobs, users - thereby being able to support both near-real-time analysis and historical traceability. Practical results show that this architecture can keep pace with the constantly evolving workload scale of high-energy physics computing environments, and has good scalability and engineering feasibility.

For the anomaly detection method, some mainstream machine learning algorithms for anomaly detection are compared according to the characteristics of high-energy physics computing workloads, where positive samples dominate and anomaly samples are few and hard to label. Finally, Isolation Forest is chosen as the anomaly detection model of choice. By choosing appropriate features and modelling, Isolation Forest can effectively recognise anomalous job behaviours without manual labelling, and its efficiency in calculations in high-dimensional and large-scale dataset environments allows it to meet practical needs.

Real production deployments validate that the proposed storage anomaly detection solution can detect abnormal storage access patterns rapidly even when the cluster is under high load with tens of thousands of concurrent jobs, and locate them accurately to specific jobs and users. This speeds up anomaly diagnosis, reduces human operation and maintenance costs, and limits the impact of abnormal job behaviour on other normal jobs. With feedback to operations personnel and the scheduling system, the overall stability and service quality of the computing platform are improved.

As a whole, this work demonstrates not only the applicability and efficacy of big data analytics and machine learning methods for storage anomaly detection in high energy physics computing systems at a technical level, but also showcases their effectiveness in real production use. The solution proposed provides an applicable and scalable technical approach to intelligent operation and maintenance of high energy physics computing platforms and useful reference for other large scale scientific computing scenarios.

ISSN: 1813-4890

References

[1] M. Ablikim, et al.: Future physics programme of BESIII, Chinese Physics C, Vol. 44 (2020) No. 4, p. 040001.

[2] F. An, et al.: Neutrino physics with JUNO, Journal of Physics G: Nuclear and Particle Physics, Vol. 43 (2016) No. 3, p. 030401.

[3] Z. Cao, et al.: The large high altitude air shower observatory (LHAASO) science book (2021 Edition), arXiv preprint arXiv:1905.02773 (2019).

[4] F. Ahmed, et al.: Centralized log management using elasticsearch, logstash and kibana, 2020 International Conference on Information Science and Communication Technology (ICISCT) (2020), p. 1-7.

[5] M. Ahmed, R. Seraj and S.M.S. Islam: The k-means algorithm: A comprehensive survey and performance evaluation, Electronics, Vol. 9 (2020) No. 8, p. 1295.

[6] J. Cervantes, et al.: A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing, Vol. 408 (2020), p. 189-215.

[7] M. Schonlau and R.Y. Zou: The random forest algorithm for statistical learning, The Stata Journal, Vol. 20 (2020) No. 1, p. 3-29.

[8] F.T. Liu, K.M. Ting and Z.H. Zhou: Isolation forest, 2008 Eighth IEEE International Conference On Data Mining (2008), p. 413-422.