Data storage technologies and artificial intelligence*

Ievgen Beliak[1,*†], Yuriy Chehil[1], Andriy Kryuchyn[1], Zichun Le[2], and Dmytro Manko[1]

[1] Institute for information Recording of NASU, Shpaka St. 2, Kyiv 03113, Ukraine
[2] Zhejiang University of Technology, Chaowang Road No.18, Hangzhou City, PR China

Abstract

A structured study of data storage technologies and their interaction with AI-systems is presented. The paper analyzes the transformation of storage architectures under the influence of increasing data volumes, intensive AI workloads, and the need for reliable long-term data preservation. The main attention is focused on the relationship between storage media, data access models, cloud-edge infrastructures, and intelligent mechanisms for data management. The article systematizes key technological trends, highlights the requirements imposed by AI-oriented applications, and outlines the main challenges related to scalability, performance, reliability, energy efficiency, and the use of AI methods for optimizing storage processes.

Keywords

data storage, cloud-edge infrastructure, AI workloads, scalability, long-term data preservation

1. Introduction

The rapid growth in the digital information volume and the spread of artificial intelligence (AI) technologies have significantly changed the requirements for data storage systems. AI-oriented applications operate with large-scale heterogeneous datasets which must be stored, processed, and reused under different performance and reliability constraints [3–5]. In this context, storage technologies can no longer be considered only as passive infrastructure for preserving information, since they directly influence the efficiency of data preparation, model training, and knowledge extraction. The development of AI systems therefore requires the coordinated evolution of storage media, cloud-edge infrastructures, hierarchical storage models, and intelligent data management mechanisms [5–7]. At the same time, the use of AI methods creates new opportunities for optimizing storage processes, including anomaly detection, predictive maintenance, and automated lifecycle management [7]. Within the framework of this study, data storage technologies are analyzed as an essential component of AI-oriented information systems, with emphasis on their technological evolution, functional requirements, and role in ensuring long-term preservation of digital data.

2. Evolution of data storage technologies for AI-oriented systems

The development of artificial intelligence systems is closely related to the long-term evolution of data storage, data transmission, and computing technologies. AI-oriented applications require not only high computational performance, but also the ability to accumulate, preserve, transfer, and repeatedly process large volumes of heterogeneous data. Therefore, the analysis of storage technologies should be considered in a broader technological context, where the density of data

*ITS-2025: Information Technologies and Security, December 11, 2025, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
beliak1312@gmail.com (Ie. Beliak); y.chehil@gmail.com (Yu. Chehil); kryuchyn@gmail.com (A. Kryuchyn); lzc@zjut.edu.cn (L. Zichun); dmitriy.manko@gmail.com (D. Manko)
0000-0001-9045-0782 (Ie. Beliak); 0009-0004-3943-7928 (Yu. Chehil); 0000-0002-5063-4146 (A. Kryuchyn); 0000-0002-9745-8565 (L. Zichun); 0000-0003-1848-2952 (D. Manko)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-26
CEUR-WS.org/Vol-4337/paper16.pdf

carriers, the bandwidth of communication channels, and the performance of computing platforms jointly determine the practical limits of AI system scalability. The first technological dimension is the evolution of storage density for different classes of data carriers. As shown in Figure 1, magne-

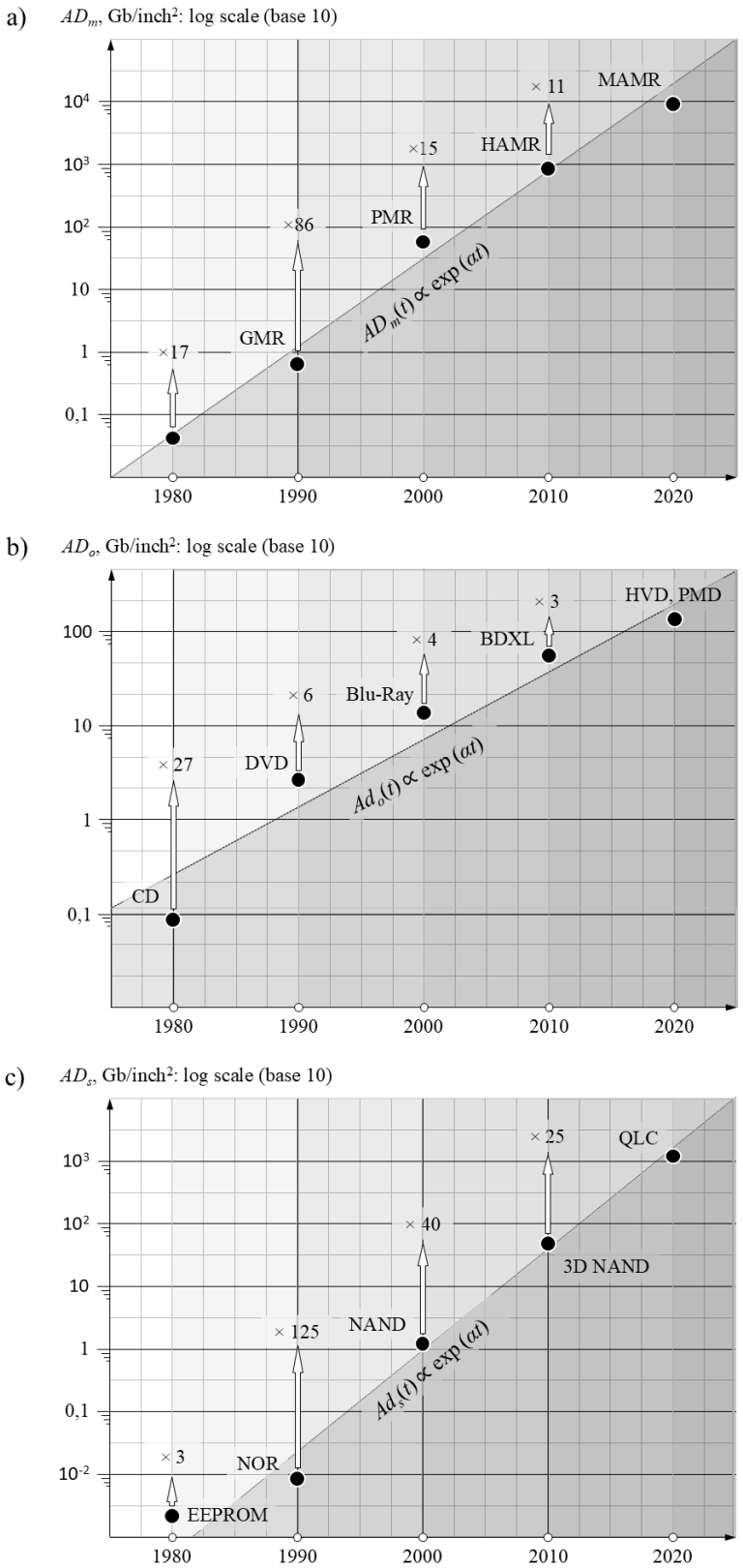


Figure 1: Evolution of storage density for different classes of data carriers: (a) magnetic recording; (b) optical storage; (c) solid-state storage.

tic, optical, and solid-state storage technologies demonstrate a long-term tendency toward exponential growth, although each class follows its own technological trajectory and application logic.

Magnetic recording evolved from early hard disk drive architectures to perpendicular magnetic recording, heat-assisted magnetic recording, and microwave-assisted magnetic recording. This evolution provided a significant increase in areal density and enabled the creation of large-capacity storage systems for data centers, cloud infrastructures, and archival platforms [1, 5, 6]. Optical storage technologies followed another trajectory, progressing from compact discs to DVD, Blu-ray, and advanced multilayer or holographic concepts. Although optical media are usually characterized by lower access speed compared with solid-state devices, they remain important in the context of long-term preservation, stability, and energy-efficient archival storage. Solid-state storage technologies, in turn, developed from early non-volatile memory concepts to NAND flash, three-dimensional NAND, and high-density QLC-based devices [18, 19]. This class of storage has become especially important for AI workloads due to its high input/output performance, low latency, and suitability for fast data ingestion and model training pipelines.

The trends summarized in Figure 1 indicate that no single class of data carrier can fully satisfy all requirements of AI-oriented information systems. Magnetic storage remains relevant for large-scale and relatively cost-efficient capacity. Optical storage is important for long-term preservation and stable archival data retention. Solid-state storage provides high-speed access to active datasets, intermediate results, checkpoints, and model artifacts. As a result, modern AI infrastructures increasingly rely on hierarchical storage architectures, where different media types are combined according to data temperature, access frequency, latency requirements, and preservation time [22].

However, the growth of storage density alone is insufficient for AI systems. The efficiency of data-intensive applications also depends on the ability to transfer data between storage, computing nodes, cloud services, and edge devices, as well as on the available computational performance. These two infrastructure factors are summarized in Figure 2.

The evolution of network bandwidth from dial-up connections and ISDN to DSL, 4G LTE, and 5G has enabled the transition from isolated computing systems to distributed and cloud-edge infrastructures [3, 4]. This is especially important for AI applications that use remote datasets, sensor streams, video data, Internet of Things devices, and geographically distributed sources of information. A similar exponential trend is observed in computing performance, as also illustrated in Figure 2. The transition from mainframes and RISC-based systems to multicore processors, hybrid GPU platforms, and exascale computing has radically increased the ability to train and deploy complex AI models. At the same time, this growth creates new pressure on storage subsystems. When computing performance increases faster than data access efficiency, storage and data movement become bottlenecks [14–17]. In practical AI pipelines, this may lead to underutilization of accelerators, delays during data loading, inefficient checkpointing, and increased energy consumption. The comparison presented in Figure 2 shows that AI-oriented systems depend on the balance between computation and data movement. High-performance processors and accelerators can only be used efficiently when storage and communication subsystems provide sufficient bandwidth, low latency, and predictable access to datasets. Therefore, the evolution of AI-oriented systems should be interpreted as the result of the joint development of three technological dimensions: storage density, communication bandwidth, and computing performance. The interaction between these dimensions explains why modern storage systems are no longer passive repositories of information.

They become active components of the AI infrastructure, responsible for fast data access, reliable preservation, efficient movement of datasets, and support for distributed processing. This also explains the growing importance of hierarchical storage, solid-state tiers, cloud and edge storage models, intelligent data placement, and automated lifecycle management.

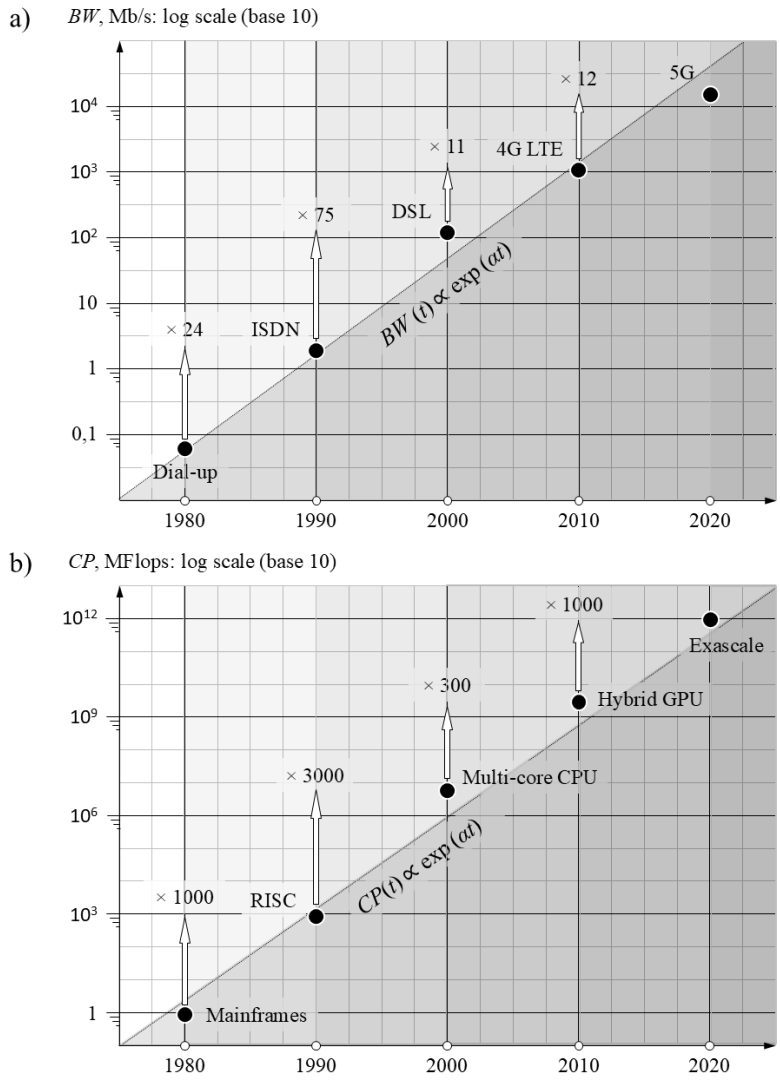


Figure 2: Evolution of infrastructure technologies supporting AI workloads: (a) network bandwidth; (b) computing performance.

The following sections further specify how the identified technological trends are transformed into practical requirements for artificial intelligence workloads. Particular attention is paid to the relationship between the type of AI task, the structure of input data, the intensity of data access, and the required storage characteristics. This makes it possible to move from a general description of technological evolution to a more applied analysis of storage architectures, data management mechanisms, and infrastructure solutions that can support training, inference, distributed processing, and long-term preservation of data in AI-oriented systems.

3. Storage requirements of artificial intelligence workloads

Artificial intelligence workloads impose specific requirements on data storage systems because they combine features of high-performance computing, big data analytics, distributed information systems, and real-time processing. In contrast to traditional enterprise applications, AI pipelines usually operate with large heterogeneous datasets that are repeatedly read, transformed, filtered, labeled, divided into training and validation subsets, and reused during model development [14–17]. As a result, the storage subsystem affects not only the preservation of data, but also the

efficiency of the entire AI lifecycle, including data ingestion, preprocessing, training, inference, monitoring, and archiving. A generalized representation of this workflow is shown in Figure 3.

As shown in Figure 3, the first stage of an AI-oriented data pipeline is associated with data acquisition and ingestion. At this stage, the storage system must support the accumulation of heterogeneous data streams, including structured records, text documents, images, video sequences, sensor measurements, log files, and experimental data. The main requirements are sufficient write bandwidth, stable operation under variable data arrival rates, and the ability to organize incoming data for further processing. In edge-oriented or distributed environments, this stage may also require local buffering, temporary caching, and delayed synchronization with cloud or central storage when network connectivity is limited [3, 20, 21].

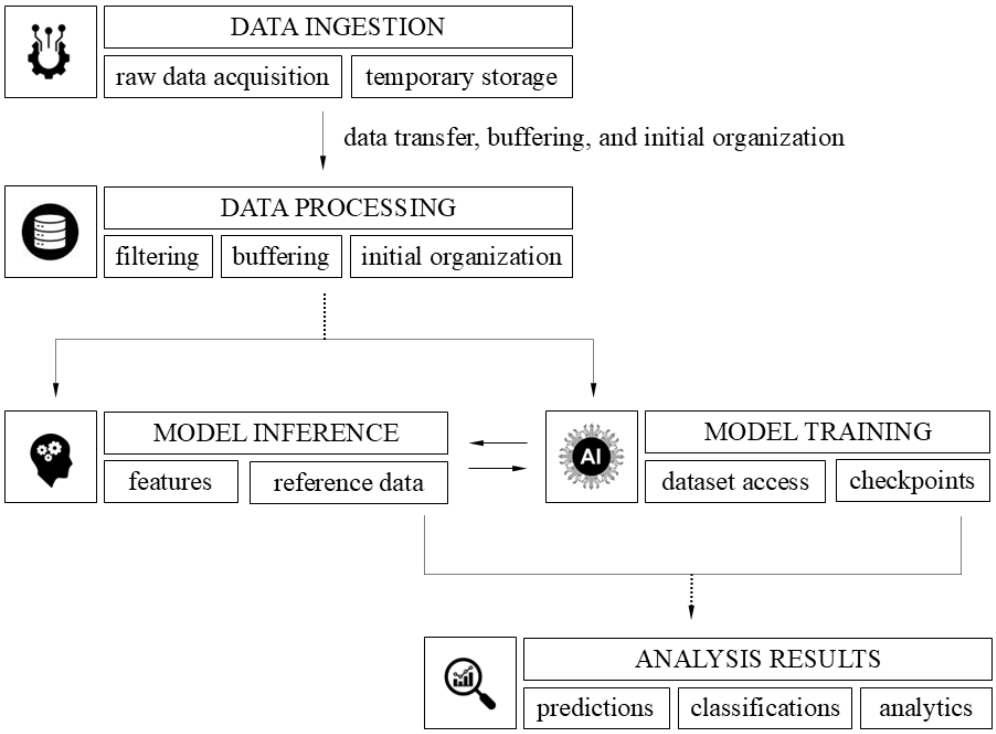


Figure 3: End-to-end workflow of AI systems from data ingestion to analytical output.

The next important requirement is efficient data preparation. Before AI models can be trained, raw data are usually cleaned, normalized, annotated, transformed, and converted into formats suitable for machine learning algorithms. These operations generate multiple intermediate datasets and metadata structures. Therefore, storage systems must provide not only capacity, but also high metadata performance, fast access to many small files, and support for dataset versioning [14–17]. This requirement is especially important for computer vision, natural language processing, and multimodal AI applications, where training collections may include millions of separate objects with different file sizes and access patterns. Model training creates one of the most intensive storage loads in AI systems. Training procedures involve repeated reading of datasets, random or semi-random sampling of minibatches, parallel access by multiple computing nodes, and periodic saving of checkpoints. If the storage subsystem cannot provide sufficient throughput and low access latency, expensive accelerators such as GPUs may remain underutilized. Therefore, training workloads require high read bandwidth, predictable latency, efficient prefetching, parallel access mechanisms, and fast storage tiers for temporary data and checkpoints. In this context, solid-state storage, NVMe-based systems, burst buffers, and distributed storage architectures become essential components of AI infrastructure [9–13]. Inference workloads form another group of requirements.

Unlike training, inference is often characterized by strict response-time constraints, especially in real-time analytics, recommendation systems, autonomous systems, industrial monitoring, and edge AI applications. In such cases, storage must support fast access to model parameters, feature stores, embeddings, reference datasets, and recent input data. Tail latency becomes particularly important, because even rare delays may reduce the quality of service. This makes low-latency storage, local caching, and optimized data placement important for production AI systems.

A separate group of requirements is related to reproducibility, governance, and long-term preservation. AI experiments often depend on specific versions of datasets, preprocessing procedures, model checkpoints, and configuration parameters. For this reason, storage systems must support data lineage, version control, access control, auditability, and reliable long-term retention of important datasets and model artifacts [6, 7]. These requirements are especially relevant for scientific, medical, industrial, and security-related applications, where the possibility of reproducing results and verifying the origin of data is critical. Finally, AI workloads require storage systems to balance performance, reliability, cost, and energy efficiency [5–8]. Active datasets and training checkpoints should be placed on high-speed storage tiers, while rarely accessed or historical data may be moved to lower-cost archival media. Such a hierarchy makes it possible to reduce infrastructure cost without losing access to valuable data. Therefore, the storage requirements of artificial intelligence workloads cannot be reduced to capacity alone. They include bandwidth, latency, metadata scalability, parallel access, fault tolerance, data lifecycle management, security, energy efficiency, and support for long-term preservation.

4. Architectures and mechanisms for AI-optimized data storage

The requirements identified in the previous section show that AI-oriented storage systems should not be limited to a single storage medium or a single access model. Modern AI pipelines combine intensive data ingestion, repeated dataset reading, checkpoint generation, inference requests, metadata operations, and long-term preservation. Therefore, storage architectures for artificial intelligence must be organized as heterogeneous and hierarchical systems, where each technological layer performs a specific function [6, 8, 22]. Such architectures combine high-speed solid-state tiers, scalable distributed storage, archival media, data lifecycle mechanisms, and intelligent control components.

One of the central elements of AI-optimized storage is the use of high-performance solid-state media. NVMe SSDs and flash-based storage arrays provide high input/output operations per second, low latency, and sufficient bandwidth for training datasets, model checkpoints, feature stores, and intermediate results [9, 10]. These characteristics are especially important when AI models are trained on GPUs or other accelerators, because insufficient data delivery may lead to idle computing resources. At the same time, solid-state storage is relatively expensive and has endurance limitations [18, 19]. For this reason, it is usually used as a hot or active tier rather than as the only storage layer in the system. A complementary architectural approach is storage disaggregation. In this case, compute and storage resources are separated and connected through high-speed network fabrics. Such organization makes it possible to scale computing nodes and storage capacity independently, improve resource utilization, and support flexible allocation of storage resources for different AI tasks. NVMe-over-Fabrics and similar technologies are especially relevant in this context because they allow remote access to high-performance storage with relatively low overhead [9, 10].

However, disaggregated architectures also increase the importance of network bandwidth, latency, congestion control, and quality-of-service mechanisms. For workloads that generate short-term intensive data flows, burst buffers and temporary high-speed storage layers are also important [11, 13]. During model training, checkpointing may produce large bursty writes that can overload shared storage. A burst buffer absorbs these peaks and later transfers data to a more durable

storage tier. Similar mechanisms are useful for preprocessing pipelines, temporary datasets, and intermediate files generated during feature extraction or model evaluation. In this way, burst-oriented storage layers help decouple computing processes from slower back-end storage. Another important direction is the use of memory-centric and near-data processing approaches. Persistent memory, computational storage, and data-path offloading mechanisms aim to reduce the cost of data movement between storage, memory, and processors [8, 12]. This is especially relevant for AI systems, where preprocessing, filtering, compression, indexing, and feature extraction may require repeated access to large datasets. If part of these operations can be performed closer to the data, the load on the host CPU, network, and main memory can be reduced.

The general logic of AI-oriented data storage is closely connected with the data lifecycle. The same dataset may exist in several forms: raw data, cleaned data, labeled samples, training subsets, validation subsets, model checkpoints, analytical outputs, and archival copies. Therefore, storage systems must support duplication, caching, migration, archiving, and metadata management. These operations should not be interpreted as secondary administrative functions. They directly influence training speed, reproducibility, storage cost, reliability, and the ability to reuse data in future AI experiments. The main lifecycle operations that should be supported by storage architectures in AI-oriented systems are summarized in Figure 4.

As shown in Figure 4, the storage layer in AI systems performs several interconnected functions. Duplication improves reliability and availability, but increases capacity requirements and requires synchronization mechanisms. Caching accelerates access to frequently used data, but depends on correct prediction of access patterns. Archiving reduces the cost of long-term preservation, but introduces delays when historical data must be restored. Data management integrates these operations through metadata, access policies, versioning, and lifecycle rules [6, 7]. In this way, the storage architecture becomes an active part of the AI infrastructure rather than a passive repository. Thus, the design of AI-oriented storage architectures requires a balance between several competing factors. High-speed storage improves training and inference performance, but increases cost and energy consumption. Archival media reduce long-term storage cost, but provide slower access. Disaggregated storage improves flexibility, but depends on network quality [9, 10].

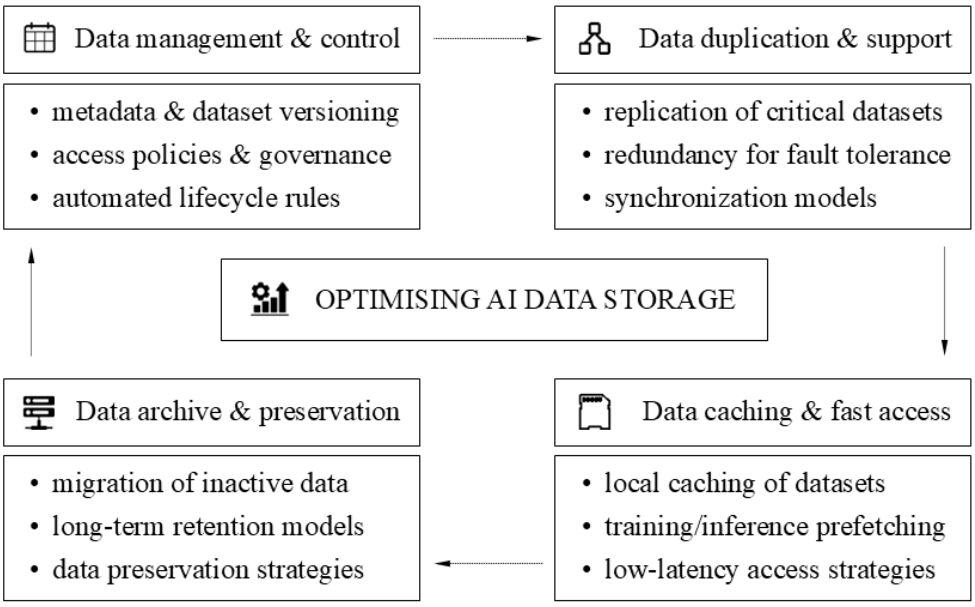


Figure 4: Core lifecycle operations in AI data storage.

Computational storage and data-path offloading reduce data movement, but require appropriate software support and standardization [8]. Therefore, the most promising approach is a hierarchical and adaptive architecture in which different storage mechanisms are selected according to data activity, workload type, performance requirements, and preservation time. This creates a basis for intelligent data management and cloud-edge storage models considered in the next section.

5. AI-assisted data management and cloud-edge storage model

The increasing complexity of AI-oriented storage infrastructures creates the need for automated and adaptive data management mechanisms. Traditional rule-based storage administration is often insufficient when workloads change dynamically, datasets are continuously updated, and processing tasks are distributed between edge devices, regional nodes, and centralized cloud or data-center resources. In this context, artificial intelligence can be used not only as a consumer of storage resources, but also as an instrument for optimizing the operation of storage systems themselves. AI-assisted data management makes it possible to analyze telemetry, predict data access patterns, detect anomalies, optimize caching, and support automated lifecycle control [7].

One of the most important directions is telemetry-driven data placement. Modern storage systems generate large volumes of operational data, including information about read and write intensity, access frequency, latency, queue depth, device temperature, error rates, network congestion, and user activity. By analyzing these parameters, AI-based models can determine which data should remain in high-speed storage tiers and which data can be moved to lower-cost or archival tiers [7]. Such placement decisions are especially important for AI workloads, where the same dataset may be intensively used during training and then become inactive after model development is completed. Adaptive data placement helps reduce storage cost while preserving fast access to active information.

Another important mechanism is self-tuning of input/output stacks. AI workloads may vary significantly depending on the type of model, dataset size, training strategy, batch size, and number of parallel workers. As a result, fixed storage parameters do not always provide optimal performance. Machine learning methods can be used to select or adjust queue depths, request sizes, caching strategies, prefetching policies, and data layout parameters [7, 15–17]. In this way, the storage subsystem can adapt to the current workload instead of relying only on predefined configurations. This is particularly relevant for distributed AI training, where inefficient data delivery may reduce accelerator utilization and increase the total time required for model training.

Predictive maintenance and anomaly detection are also essential components of AI-assisted storage management [7, 18, 19]. Storage devices and distributed storage services may fail due to wear, overheating, firmware errors, network interruptions, metadata inconsistencies, or unexpected workload peaks. AI-based monitoring can detect early signs of degradation and identify abnormal behavior before it leads to data loss or service interruption. For example, predictive models can estimate the probability of device failure, identify unusual latency growth, or detect inconsistencies in replication and synchronization processes. This allows the system to initiate preventive actions, such as data migration, replication, replacement of risky devices, or load redistribution.

AI-assisted data reduction is another promising direction [7, 8]. Large datasets used in artificial intelligence often contain duplicated, redundant, or rarely accessed information. Intelligent compression, deduplication, filtering, and compaction policies can reduce storage consumption and improve data transfer efficiency. However, data reduction should be applied carefully, because it may increase computational overhead or affect the speed of access to training data. AI-based control mechanisms can help select an appropriate trade-off between storage savings, processing cost, access latency, and energy consumption. These mechanisms are especially important in cloud-edge storage models [4, 20, 21]. In such systems, data are not stored and processed in a single

location. Edge nodes collect data near their sources, perform preliminary filtering or inference, and temporarily store information under limited resource and connectivity conditions. Regional aggregation layers can consolidate, compress, and preprocess data received from multiple edge nodes. Centralized cloud or data-center infrastructures provide large-scale storage, model training, long-term archiving, and global coordination. Therefore, the storage architecture must support continuous movement of data between local, regional, and central levels. In a cloud-edge storage model, the edge ingestion tier is responsible for receiving data from sensors, devices, cameras, industrial systems, or user applications. Its main function is to provide fast local recording, short-term buffering, and support for low-latency inference. The regional aggregation layer performs intermediate processing, filtering, and synchronization. It can reduce the amount of data transferred to the cloud by selecting only relevant fragments or compressed representations. The core training cluster uses high-performance storage resources for repeated access to large datasets, checkpoint generation, and model optimization [11, 13–17]. Durable shared storage preserves validated datasets and model artifacts, while the long-term archive tier supports reproducibility, compliance, and historical data retention.

The integration of AI-assisted management into this cloud-edge model creates a closed-loop control system. At the first stage, the storage infrastructure observes telemetry from devices, networks, applications, and user activity. At the second stage, analytical models infer the current state of the system and predict future workload changes. At the third stage, the control plane selects appropriate actions, such as data migration, cache update, replication adjustment, anomaly response, or archival transfer. At the final stage, these actions are applied with predefined constraints, security policies, and human supervision when necessary. Such a loop allows the storage infrastructure to react to workload changes and gradually improve the efficiency of data management.

Security and governance must remain integrated into all levels of this model. AI-oriented storage systems often process sensitive, scientific, industrial, or personal data. Therefore, automated data movement should be combined with access control, encryption, audit logs, metadata tracking, and policy-based restrictions. In distributed cloud-edge environments, this is especially important because data may be temporarily stored outside centralized infrastructure. The system must ensure that optimization does not violate security requirements, data ownership rules, or reproducibility conditions.

Thus, AI-assisted data management and cloud-edge storage should be considered as complementary components of modern AI infrastructure. Cloud-edge organization distributes data and processing according to latency, bandwidth, and resource constraints, while AI-assisted management provides adaptive control over placement, caching, reliability, and lifecycle operations. Together, these mechanisms make it possible to build storage systems that are not only scalable and high-performance, but also self-monitoring, cost-aware, and suitable for long-term operation under changing AI workloads.

6. Discussion: challenges and research directions

The analysis of data storage technologies for artificial intelligence systems shows that the main challenge is not limited to increasing storage capacity. Modern AI infrastructures require a balanced combination of performance, scalability, reliability, cost efficiency, energy efficiency, and data governance. Therefore, the evaluation of storage solutions should be performed not only by the amount of stored data, but also by their ability to support intensive data access, distributed processing, reproducibility of experiments, and long-term preservation of information.

One of the key challenges is metadata scalability [6, 14, 15]. AI pipelines generate not only large datasets, but also a significant amount of auxiliary information, including labels, annotations, preprocessing parameters, dataset versions, model checkpoints, logs, and experiment

configurations. In large-scale systems, metadata operations may become a bottleneck even when the storage media provide sufficient bandwidth. This problem is especially important for machine learning workflows that require frequent access to many files, rapid search through dataset versions, and reliable tracking of data lineage. Therefore, future storage architectures should include scalable metadata services and efficient mechanisms for managing dataset provenance. Another important issue is the processing of small-file and multimodal datasets [14–17]. Many AI applications, especially in computer vision, natural language processing, medical diagnostics, remote sensing, and sensor analytics, operate with millions of relatively small objects. Such workloads differ from traditional sequential access patterns and may overload file systems, indexing mechanisms, and storage controllers. At the same time, multimodal datasets combine images, video fragments, text documents, structured records, audio signals, and sensor measurements. This increases the complexity of data organization and requires storage systems that can efficiently support heterogeneous formats, access patterns, and metadata structures. Quality of service is also a critical research direction [9, 10]. In multi-tenant AI infrastructures, several users or applications may simultaneously perform data ingestion, model training, inference, checkpointing, and archival operations. Without appropriate quality-of-service mechanisms, intensive training workloads may negatively affect inference latency or delay access to shared datasets. This problem becomes even more significant in disaggregated storage and cloud-edge environments, where data access depends on network conditions. Therefore, future systems should provide workload-aware scheduling, priority-based access, bandwidth isolation, and mechanisms for controlling tail latency. Edge synchronization remains another unresolved problem [20, 21]. Cloud-edge AI systems often operate under conditions of limited bandwidth, intermittent connectivity, and heterogeneous local resources. Edge nodes may collect and process data locally, while central cloud or data-center infrastructures are responsible for long-term storage, large-scale training, and global coordination. This creates challenges related to delayed synchronization, conflict resolution, data consistency, version control, and selective transfer of only relevant data fragments. Future research should focus on adaptive synchronization strategies that consider data importance, freshness, privacy restrictions, and available communication resources. Reliability of solid-state storage is also important for AI-oriented infrastructures [18, 19]. Flash-based and NVMe storage devices provide high performance, but their operation is affected by endurance limits, wear processes, temperature, workload intensity, and controller behavior. Since AI workloads may generate intensive read and write operations, including repeated access to datasets and frequent checkpointing, reliability-aware storage management becomes necessary. Predictive maintenance, anomaly detection, data migration from risky devices, and adaptive placement of write-intensive workloads are promising directions for increasing the robustness of AI storage systems. Computational storage and near-data processing represent another promising but still developing research direction [8]. The possibility of performing filtering, compression, indexing, or preprocessing closer to the data can reduce data movement and improve the efficiency of AI pipelines. However, practical implementation of this approach requires standard interfaces, secure execution mechanisms, programmability, and integration with existing software stacks. Without these components, computational storage may remain limited to specialized scenarios rather than becoming a general element of AI infrastructure. The evaluation methodology for AI-oriented storage systems should also be improved [14–17, 22]. Traditional benchmarks based only on bandwidth or input/output operations per second are insufficient for characterizing real AI workflows. More representative evaluation should include end-to-end training time, accelerator utilization, inference latency, checkpoint overhead, metadata operation rate, data restoration time, energy consumption, cost per processed dataset, and reproducibility support. Such metrics make it possible to compare storage architectures not only as hardware platforms, but also as components of the full AI data lifecycle.

Thus, the main research directions in AI-oriented data storage include scalable metadata management, optimization for small-file and multimodal datasets, quality-of-service mechanisms, reliable cloud-edge synchronization, predictive maintenance of high-performance storage, practical use of computational storage, and development of workload-aware evaluation metrics. These challenges confirm that storage technologies should be considered as an active and intelligent layer of AI infrastructure. Their further development will directly influence the scalability, reliability, and practical efficiency of artificial intelligence systems.

Conclusions

The conducted analysis shows that the development of artificial intelligence systems is inseparable from the evolution of data storage technologies and related infrastructure components. The growth of storage density, communication bandwidth, and computing performance has created the technological basis for large-scale AI workloads, but at the same time has increased the requirements for data access, reliability, scalability, and lifecycle management. In this context, storage systems can no longer be regarded only as passive repositories of digital information. They become an active part of AI-oriented infrastructures, influencing the efficiency of data ingestion, preprocessing, model training, inference, reproducibility of experiments, and long-term preservation of valuable datasets. The paper substantiates the need for hierarchical and adaptive storage architectures that combine high-speed solid-state tiers, distributed and disaggregated storage, caching, archiving, and near-data processing mechanisms. It is also shown that artificial intelligence methods can be used not only as consumers of storage resources, but also as tools for telemetry-based data placement, self-tuning of input/output processes, anomaly detection, predictive maintenance, and cloud-edge coordination.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, based on GPT-5.5 Thinking, in order to: grammar and spelling check, and language editing of author-provided materials. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop Distributed File System, in: 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), IEEE, 2010, pp. 1–10. doi:10.1109/MSST.2010.5496972.

[2] D. Gudu, M. Hardt, A. Streit, Evaluating the performance and scalability of the Ceph distributed storage system, in: 2014 IEEE International Conference on Big Data (Big Data), IEEE, 2014, pp. 177–182. doi:10.1109/BigData.2014.7004229.

[3] W. Shi, J. Cao, Q. Zhang, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (2016) 637–646. doi:10.1109/JIOT.2016.2579198.

[4] H. Cai, B. Xu, L. Jiang, A. V. Vasilakos, IoT-based big data storage systems in cloud computing: Perspectives and challenges, IEEE Internet of Things Journal 4 (2017) 75–87. doi:10.1109/JIOT.2016.2619369.

[5] S. Mazumdar, et al., A survey on data storage and placement methodologies for cloud-big data ecosystem, Journal of Big Data 6 (2019) 65. doi:10.1186/s40537-019-0178-3.

[6] R. Macedo, J. Paulo, J. Pereira, A. Bessani, A survey and classification of software-defined storage systems, ACM Computing Surveys 53 (3, 2020) Article 48. doi:10.1145/3385896.

[7] Y. Liu, H. Wang, K. Zhou, C. Li, R. Wu, A survey on AI for storage: Architecture-oriented, data-specific, operation & maintenance, CCF Transactions on High Performance Computing (2022). doi:10.1007/s42514-022-00101-3.

[8] D. Fakhry, M. Abdelsalam, M. W. El-Kharashi, M. Safar, A review on computational storage devices and near memory computing for high performance applications, Memories - Materials, Devices, Circuits and Systems 4 (2023) 100051. doi:10.1016/j.memori.2023.100051.

[9] Z. Guz, A. Shayesteh, M. Balakrishnan, NVMe-over-Fabrics performance characterization and the path to low-overhead storage disaggregation, ACM Transactions on Storage (2018). doi:10.1145/3258045.

[10] Z. Jia, et al., When NVMe over Fabrics meets Arm: Performance and implications, in: 2019 35th Symposium on Mass Storage Systems and Technologies (MSST), IEEE, 2019. doi:10.1109/MSST.2019.00009.

[11] R. F. da Silva, et al., On the use of burst buffers for accelerating data-intensive workflows, in: Proceedings of the 2017 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC), ACM, 2017. doi:10.1145/3150994.3151000.

[12] M. Weiland, et al., An early evaluation of Intel's Optane DC persistent memory module and its impact on high-performance scientific applications, in: Proceedings of SC'19, ACM, 2019. doi:10.1145/3295500.3356159.

[13] M. A. Vef, et al., GekkoFS: A temporary burst buffer file system for HPC applications, Journal of Computer Science and Technology 35 (2020) Article 1. doi:10.1007/s11390-020-9797-6.

[14] J. L. Bez, et al., I/O Access Patterns in HPC Applications: A 360-Degree Survey, ACM Computing Surveys (2023). doi:10.1145/3611007.

[15] S. Pumma, et al., Scalable deep learning via I/O analysis and optimization, ACM Transactions on Parallel Computing 6 (2, 2019) Article 6. doi:10.1145/3331526.

[16] A. Aizman, G. Maltby, T. Breuel, High performance I/O for large scale deep learning, in: 2019 IEEE International Conference on Big Data (BigData), IEEE, 2019. doi:10.1109/BigData47090.2019.9005703.

[17] J. Huang, et al., HVAC: Removing I/O bottleneck for large-scale deep learning applications, in: 2022 IEEE International Conference on Cluster Computing (CLUSTER), IEEE, 2022. doi:10.1109/CLUSTER51413.2022.00044.

[18] J. Meza, Q. Wu, S. Kumar, O. Mutlu, A large-scale study of flash memory failures in the field, in: Proceedings of the ACM SIGMETRICS, ACM, 2015, pp. 177–190. doi:10.1145/2796314.2745848.

[19] A. S. Spinelli, C. Monzio Compagnoni, A. L. Lacaita, Reliability of NAND flash memories: Planar cells and emerging issues in 3D devices, Computers 6 (2017) Article 16. doi:10.3390/computers6020016.

[20] H. Jin, R. Luo, Q. He, S. Wu, Z. Zeng, X. Xia, Cost-effective data placement in edge storage systems with erasure code, IEEE Transactions on Services Computing 16 (2022) 1039–1050. doi:10.1109/TSC.2022.3152849.

[21] X. Xia, et al., Popularity-aware data placement in erasure coding-based edge storage systems, IEEE Transactions on Parallel and Distributed Systems 36 (2025) 2733–2746. doi:10.1109/TPDS.2025.3619273.

[22] G. Zhang, et al., Survey of storage systems in high performance computing, CCF Transactions on High Performance Computing (2025). doi:10.1007/s42514-025-00268-5.