

Carbon-aware Storage Tiering Framework

Nedasadat Taheri^{1,*}, Seyed Hossein Taheri², Houman Kosarirad¹

¹University of Nebraska-Lincoln

²Islamic Azad University, Science and Research Branch

Abstract

The rapid growth of cloud services and serverless applications has significantly increased the demand for efficient storage management in modern data centers. Tiered storage architectures, which combine high-performance solid-state drives (SSDs) with high-capacity hard disk drives (HDDs), are commonly used to balance performance and cost. However, accurately identifying which data objects should reside in fast storage tiers remains a challenging task due to dynamic and skewed access patterns in real-world workloads. In response, this paper introduces a predictive and carbon-aware storage management framework designed to improve data placement decisions in tiered storage systems. Our approach analyzes blob access traces and employs workload prediction to estimate future data hotness, enabling proactive migration of frequently accessed objects to high-performance storage tiers. By integrating prediction with migration cost awareness and time-varying grid carbon-intensity signals, the framework defers eligible migrations during carbon-intensive periods and prioritizes them during lower-carbon periods. Evaluation on an activity-focused subset of the Azure Functions Blob Access Trace, together with larger-scale experiments using up to 1,000 blobs, shows that the carbon-aware policy reduces operational carbon emissions by 27.76% relative to an otherwise equivalent LSTM energy-aware policy, while reducing total energy consumption by 24.70% and the number of migrations by 50%. A constant-carbon-intensity counterfactual further shows that temporal carbon-aware scheduling contributes an additional 3.05 percentage points of emissions reduction beyond the reduction attributable to lower energy consumption alone. The migration-energy-weighted carbon intensity is reduced from 269.66 to 174.65 gCO₂e/kWh, while average access latency remains essentially unchanged. These results demonstrate that combining workload prediction, migration-cost awareness, and temporally varying carbon signals can improve the environmental efficiency of tiered storage management without compromising system performance.

Received on 17 August 2026, accepted on 24 September 2026, published on 08 October 2026

Keywords: Tiered storage, cloud storage, workload prediction, hot data management, energy-efficient storage, carbon-aware computing, data migration

Copyright © 2026 Nedasadat Taheri *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/

1. Introduction

Modern cloud computing platforms rely heavily on large-scale storage systems to support data-intensive applications. With the rapid growth of serverless computing and cloud services, storage infrastructures must handle massive volumes of data while maintaining low latency, high availability, and energy efficiency. In many cloud environments, storage systems adopt a *tiered architecture*, where frequently accessed data is placed on high-performance devices

such as SSDs, while less frequently accessed data is stored on lower-cost and higher-capacity devices such as HDDs.

A key challenge in tiered storage systems is identifying which data objects should reside in the fast storage tier. Access patterns in cloud workloads are often highly skewed and dynamic, where a small fraction of objects (referred to as *hot data*) account for a large portion of total accesses. Efficiently identifying and placing hot data on fast storage tiers can significantly improve system performance and reduce latency.

*Corresponding author. Email: ntaheri2@huskers.unl.edu

Traditional storage management systems often rely on simple heuristics such as recent access frequency or threshold-based policies to determine storage placement. Although these approaches are computationally inexpensive, they often fail to capture the temporal dynamics of real-world workloads. As a result, such methods may either trigger unnecessary data migrations or fail to proactively move data before demand spikes occur.

At the same time, energy consumption and environmental sustainability have become critical concerns for modern data centers. Storage systems contribute significantly to the overall energy footprint of cloud infrastructures. Frequent data migrations, inefficient storage placement, and unnecessary disk accesses can lead to increased power consumption and operational carbon emissions. Consequently, there is a growing interest in *energy-aware* and *carbon-aware* storage management strategies that aim to optimize both system performance and environmental impact.

Recent studies have explored predictive workload modeling and carbon-aware computing independently. However, few works integrate workload prediction, storage tier management, and energy-aware decision policies into a unified framework. In particular, existing approaches often rely on static heuristics and do not fully leverage predictive models to anticipate future access patterns.

Moreover, evaluating carbon-aware storage management using a constant carbon-intensity assumption cannot capture temporal variations in the electricity grid and therefore cannot demonstrate whether storage operations are actually shifted toward cleaner periods. To address this issue, our framework incorporates a real historical time-varying grid carbon-intensity trace, temporally aligned with the storage workload. Forecast carbon intensity is used when making migration decisions, while the corresponding actual carbon intensity is used to evaluate the realized operational emissions.

To better capture the environmental impact of storage operations, we estimate carbon emissions using the standard relationship between energy consumption and grid carbon intensity. Unlike a constant-carbon-intensity evaluation, the carbon intensity in our revised framework varies hourly, allowing the environmental impact of both storage accesses and migration operations to depend on when those operations occur.

In this paper, we propose a predictive and carbon-aware storage migration framework that combines workload forecasting with energy-aware storage placement decisions. Using access traces from the Azure Functions Blob Access dataset, our system predicts future blob access patterns and proactively migrates likely hot data objects to high-performance storage

tiers. The proposed framework also incorporates migration cost estimation to avoid unnecessary data movement and reduce energy consumption. In addition, the migration controller uses forecast grid carbon intensity to defer eligible migrations during carbon-intensive periods and permit placement updates during lower-carbon periods. This enables the framework to jointly consider predicted data hotness, migration cost, and the temporal carbon intensity of electricity.

The primary experimental configuration focuses on the 200 most frequently accessed blobs rather than attempting to model the entire long tail of the trace. Although these 200 objects represent only 0.0435% of the 459,580 unique blobs observed in the analyzed workload, they account for 65.02% of the 2,000,000 access events. The subset is therefore intended as an activity-focused workload that preserves the dominant access concentration and temporal behavior relevant to hot-data placement, rather than as a statistically representative sample of every workload characteristic. In particular, its read/write composition differs from that of the complete trace, and this difference is considered when interpreting the results.

The main contributions of this paper are summarized as follows:

- We analyze a real-world cloud workload dataset and characterize its strongly skewed hot-cold access behavior. The primary evaluation focuses on the 200 most frequently accessed blobs, which account for 65.02% of the 2,000,000 access events in the analyzed trace despite representing only 0.0435% of its unique blobs.
- We design an LSTM-based predictive hotness estimation model using a 24-hour historical access window to forecast subsequent hourly access intensity and dynamically identify the top 10% of blobs as hot-data candidates. On the held-out test period, the model achieves an R^2 of 0.9197; the predicted Top-20 set captures 99.92% of actual hourly accesses and achieves a mean NDCG@20 of 0.9984.
- To examine whether the prediction and ranking behavior depends on the 200-blob experimental scale, we additionally evaluate the pipeline using the 500 and 1,000 most frequently accessed blobs. The corresponding access matrices contain 500×336 and 1,000×336 blob-hour observations, compared with 200×336 in the primary configuration. Across the 200-, 500-, and 1,000-blob configurations, the predicted top-10% sets capture 99.92%, 99.98%, and 99.97% of hourly accesses on average, respectively. Mean NDCG@ k remains similarly stable at 0.9984, 0.9987, and 0.9987,

indicating that the ranking quality remains robust as the evaluated object population increases.

- We propose a carbon-aware storage migration policy that jointly considers predicted data hotness, size-dependent migration energy, and time-varying grid carbon intensity. Migration decisions use forecast carbon intensity, whereas realized emissions are calculated using actual hourly carbon intensity.
- We implement a simulation framework that evaluates Always-HDD, Always-SSD, LSTM Energy-Aware, and LSTM Carbon-Aware storage policies under the same workload, device-energy, migration-cost, and latency models.
- Experimental results show that the LSTM Carbon-Aware policy reduces operational carbon emissions by 27.76%, total energy consumption by 24.70%, and migration frequency by 50% relative to the otherwise equivalent LSTM Energy-Aware policy, while average access latency remains essentially unchanged.
- We further separate energy-efficiency gains from temporal carbon-aware scheduling through a constant-carbon-intensity counterfactual. The analysis shows that time-varying carbon awareness contributes an additional 3.05 percentage points of emissions reduction beyond the reduction attributable to lower energy consumption alone, while reducing migration-energy-weighted carbon intensity from 269.66 to 174.65 gCO₂e/kWh. A threshold sensitivity analysis across the 25th–75th percentiles of the training carbon-intensity distribution yields carbon reductions of 27.66%–27.79%, demonstrating robustness to the selected carbon threshold.

The remainder of this paper is organized as follows. Section Related Work reviews related work on hot-cold data classification, energy-aware storage systems, and carbon-aware computing. Section Methodology presents the proposed predictive migration framework. Section Experimental Evaluation describes the experimental setup and evaluation results. Section Discussion discusses the implications of the findings, and Section Conclusion concludes the paper.

2. Related Work

Efficient storage management in large-scale cloud systems has been widely studied from multiple perspectives, including hot-cold data classification, hierarchical storage architectures, energy-aware storage systems, and carbon-aware computing frameworks. In

this section, we review the most relevant research directions that motivate our proposed approach.

2.1. Hot and Cold Data Classification

One of the fundamental concepts in large-scale storage systems is the classification of data based on access frequency. Frequently accessed objects are considered *hot data* and are typically stored on high-performance storage tiers such as SSDs, while infrequently accessed objects are treated as *cold data* and migrated to slower but more energy-efficient storage devices such as HDDs.

Han and Song introduced a QoE-aware storage power management framework in [1] that identifies hot video segments based on popularity metrics and quality-of-experience impact. Their approach formulates the storage allocation problem as a knapsack optimization problem to maximize user experience while reducing disk power consumption. Their evaluation demonstrated that controlled QoE degradation could reduce disk power consumption by up to 46%.

Similarly, Guan et al. proposed an optimization strategy for Hadoop Distributed File System (HDFS) in [2]. Their Smart Storage Management (SSM) framework dynamically classifies files according to access frequency and automatically migrates them between SSD and HDD tiers using a rule-based management system. The architecture incorporates multiple modules such as a StatesManager and RuleManager to track file access patterns and trigger storage migrations dynamically.

Recent work by Ma et al. [3] introduced a data temperature model that uses access recency and frequency to determine storage placement. Their system further integrates Long Short-Term Memory (LSTM) networks to predict future access patterns and perform proactive data migration between storage tiers.

2.2. Energy-Aware Storage Systems

Energy consumption is a major operational cost for modern data centers. Several studies have proposed energy-aware storage architectures that dynamically migrate data based on workload characteristics.

Xie proposed the MQHD framework in [4]. The system introduces a multi-queue hot data recognition mechanism and a Heat Degree (HD) metric to identify heavily accessed data blocks. The framework also introduces a Load Pressure Ratio (LPR) to determine when storage load balancing is required, enabling dynamic migration of hot data to prevent disk overload while maintaining energy efficiency.

Zhizhuo et al. proposed an energy-efficient storage architecture for video surveillance systems in [5]. Their Semi-RAID architecture concentrates disk activity on a subset of drives while allowing other disks to remain idle, thereby reducing overall energy consumption.

Similarly, Darwich et al. [6] proposed a hierarchical storage system that categorizes video segments according to popularity. Frequently accessed video segments are stored in high-speed storage, while less popular content is migrated to lower-cost archival storage.

These studies demonstrate that dynamic data migration and workload-aware storage placement can significantly improve storage efficiency and reduce energy consumption in large-scale storage systems. Prior work has also examined the differences in energy consumption among storage technologies and the relationship between I/O workloads and storage-device energy consumption [12, 13]. These studies provide a basis for analytically modeling the distinct performance and energy characteristics of SSD and HDD storage devices.

2.3. Carbon-Aware Computing

More recently, researchers have begun exploring carbon-aware computing approaches that incorporate power grid carbon intensity into system-level decision making.

Raut et al. proposed a framework in [7] that evaluates storage placement decisions based on carbon intensity, power usage effectiveness (PUE), and energy consumption metrics. Their model estimates carbon emissions using the relationship between workload energy consumption and grid carbon intensity.

Köhler et al. [8] introduced a system called cMemento that dynamically migrates data between heterogeneous memory tiers based on carbon intensity and energy efficiency. Their system demonstrates that carbon-aware memory placement can reduce operational emissions by up to 23%.

Gebhard [9] proposed a scheduling framework that delays non-critical system operations until periods of lower grid carbon intensity. The approach integrates carbon-intensity information with system scheduling to shift flexible tasks toward environmentally preferable time windows.

In addition, Rao and Chien [10] investigate the operational carbon footprint associated with storage reliability and management activities, highlighting opportunities for carbon-aware management of storage-system operations.

Collectively, these studies highlight an important distinction between merely estimating emissions from energy consumption and actively using time-varying carbon-intensity information to influence system decisions. In particular, temporal carbon-aware scheduling can exploit variations in grid carbon intensity by shifting flexible operations toward lower-carbon periods. This motivates incorporating a temporally varying carbon signal directly into storage migration decisions rather than evaluating storage policies under a constant carbon-intensity assumption.

2.4. Carbon Modeling and Sustainable Data Center Design

Beyond storage systems, prior research has emphasized the importance of accounting for the broader carbon footprint of computing infrastructure. Gupta et al. [11] demonstrate that evaluating the environmental impact of computing requires consideration of both operational emissions and the embodied carbon associated with hardware manufacturing.

For the operational component considered in this study, energy consumed by storage accesses and data migration must be translated into carbon emissions according to the carbon intensity of the electricity used at the corresponding time. In addition, data movement itself introduces energy overhead. Prior analytical work has modeled the energy consumption associated with network transfers and migration as a function of transfer duration and power consumption [14]. These observations motivate explicitly accounting for both storage-device activity and network-transfer energy when estimating the operational carbon consequences of storage migration.

Although prior work has separately advanced predictive storage tiering and carbon-aware computing, an important technical gap remains between these two research directions. Existing predictive tiering approaches primarily use historical or forecast workload behavior to determine *which* objects should reside in faster storage tiers [1–3]. Their placement decisions are generally driven by access frequency, recency, predicted data temperature, performance, or storage-energy considerations, but the timing of the resulting data migrations is not explicitly conditioned on forecast time-varying grid carbon intensity. Consequently, a migration that is desirable from a workload or performance perspective may still be executed during a carbon-intensive grid interval.

Conversely, existing carbon-aware computing approaches primarily address *when* flexible computation or background system operations should execute by exploiting temporal or spatial variations in grid carbon intensity [7–10]. Carbon-aware storage and memory studies have demonstrated the value of shifting flexible operations toward cleaner electricity periods, but they generally do not combine this temporal decision with next-interval object-level hotness prediction for SSD–HDD placement. Thus, carbon-aware scheduling alone does not determine which storage objects are most valuable to promote or demote based on anticipated near-term access demand.

The technical distinction of the proposed framework is therefore the explicit coupling of these two decisions. First, an LSTM predicts the next-hour access intensity of each blob, and the predicted values are ranked to

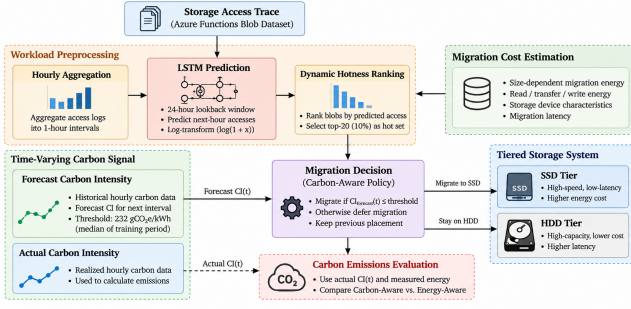


Figure 1. Architecture of the proposed predictive carbon-aware storage tiering framework

construct a dynamic Top- k target SSD set. Second, changes between the current and desired placements are translated into size-dependent migration costs that account for source reads, network transfer, and destination writes. Third, forecast grid carbon intensity is used as a temporal execution constraint: eligible placement changes are executed during lower-carbon intervals and deferred during forecast high-carbon intervals. Actual carbon intensity is kept separate from the controller and is used only for retrospective emissions accounting.

Accordingly, the contribution is not simply the simultaneous use of workload prediction and carbon information. Rather, the framework connects object-level predictive tiering with carbon-aware migration timing under a common SSD–HDD placement model. This enables the evaluation to distinguish the benefit of predicting *which* objects should occupy the SSD tier from the additional benefit of deciding *when* the corresponding placement changes should be executed.

3. Methodology

This section describes the architecture and workflow of the proposed predictive carbon-aware storage tiering framework (Fig. 1).

3.1. System Overview

The system operates on storage access traces and periodically performs data placement decisions across heterogeneous storage tiers.

Let $\mathcal{B} = \{b_1, b_2, \dots, b_n\}$ denote the set of storage objects (blobs) observed in the workload trace. Each blob b_i is associated with an access time series

$$A_i = \{a_i^1, a_i^2, \dots, a_i^T\}$$

where a_i^t represents the number of accesses to blob b_i during time interval t .

The system operates in discrete decision intervals and performs the following five stages:

1. Workload preprocessing

Raw storage access logs are aggregated into fixed time intervals to construct access sequences for each blob. Each sequence is used as the input representation for the prediction model. In our implementation, accesses are aggregated at an hourly resolution. A 24-hour lookback window is used to predict the access intensity of each blob during the subsequent hour.

2. Hotness prediction

A Long Short-Term Memory (LSTM) network (Table 1) is used to estimate the future access intensity of each blob. Given the past L observations

$$\{a_i^{t-L+1}, \dots, a_i^t\}$$

the model predicts the next access value

$$\hat{a}_i^{t+1}$$

which represents the expected demand for the next decision interval.

Because the access-count distribution is highly skewed, the hourly access counts are transformed using $\log(1 + a_i^t)$ before model training. The predicted values are subsequently transformed back to the original access-count scale for evaluation and storage-placement decisions.

The primary prediction experiment is conducted on the 200 most frequently accessed blobs. This subset was selected to concentrate the evaluation on the active portion of the workload that drives storage-tiering decisions. Although these 200 blobs represent only 0.0435% of the 459,580 unique blobs observed in the analyzed trace, they account for 65.02% of the 2,000,000 access events. The resulting 200-blob access matrix contains 336 hourly intervals and is highly sparse, with 93.41% of the blob-hour entries containing no accesses. Thus, the selected workload retains the strong access concentration and temporal sparsity that motivate predictive hot-data placement. However, because selection is based on total access frequency, the subset is intentionally activity-focused rather than a statistically representative sample of the complete trace; in particular, its read/write composition is more read-dominated than that of the full workload.

The modeled access latencies of 0.2 ms for SSD and 5 ms for HDD are representative values used to capture the substantial latency difference between solid-state and rotational storage. They are not intended to characterize a specific

Table 1. LSTM architecture and training configuration

Parameter	Value
Input sequence length	24 hourly observations
Prediction horizon	1 hour
LSTM layers	1
LSTM hidden units	64
LSTM activation	tanh
Recurrent activation	sigmoid
Dropout rate	0.20
Dropout placement	After LSTM layer
Dense hidden units	32
Dense activation Output units	ReLU
Output activation	1
Optimizer	Linear
Learning rate	Adam
Loss function	0.001
Batch size	MSE
Maximum epochs Early-stopping monitor Early-stopping patience	128
Restore best weights	40
Temporal split	Validation loss
Input transformation	5 epochs
Primary training epochs	Yes
Primary training time	70/15/15
	$\log(1 + x)$
	12
	165.37 s

commercial device. Similarly, the bandwidth and power parameters define a controlled analytical configuration for estimating relative access and migration costs. Prior studies have demonstrated that SSD and HDD technologies exhibit substantially different response-time, throughput, and energy characteristics and that both device power and operation duration must be considered when estimating storage energy consumption [12, 13].

Each input sample consists of 24 consecutive hourly access counts for a blob, and the prediction target is its access intensity in the immediately following hour. Because the access-count distribution is highly skewed, the input counts are transformed using $\log(1 + x)$ before model training. Predictions are transformed back to the original access-count scale before prediction evaluation and storage-placement decisions.

The network consists of a single LSTM layer with 64 hidden units using tanh activation and sigmoid recurrent activation. A dropout layer with a rate of 0.20 is applied to the LSTM output, followed by a fully connected layer with 32 units and ReLU activation. The final regression layer contains one unit with linear activation. The model is optimized using Adam with an initial learning rate of 0.001 and mean squared error (MSE) as the

loss function. Training uses a batch size of 128 and a maximum of 40 epochs. Early stopping monitors the validation loss with a patience of five epochs and zero minimum improvement threshold; the weights corresponding to the best validation loss are restored before evaluation. In the primary 200-blob experiment, training terminated after 12 epochs and required approximately 165.37 seconds.

The experiments were implemented in Python 3.13.15 using TensorFlow 2.20.0, NumPy 2.1.3, and Pandas 2.2.3 under a 64-bit Linux x86_64 environment. TensorFlow did not detect or use a GPU during the reported execution; therefore, model training and the trace-driven simulations were executed on the CPU. The measured training times for the 500- and 1,000-blob scalability experiments were approximately 404.50 and 514.29 seconds, respectively.

To examine whether the conclusions drawn from the 200-blob configuration are sensitive to workload scale, we additionally construct 500- and 1,000-blob configurations using the same selection and preprocessing procedure. These configurations produce access matrices of 500×336 and 1000×336 , respectively, compared with 200×336 for the primary experiment. The corresponding matrix sparsities increase from 93.41% for 200 blobs to 96.76% and 97.92% for 500 and 1,000 blobs, reflecting the increasingly sparse long tail of the workload.

Separate LSTM models are trained for each workload scale using the same 24-hour lookback formulation and temporal evaluation procedure. To evaluate the ranking behavior that directly drives storage placement, we measure the fraction of hourly accesses captured by the predicted top 10% of blobs and the normalized discounted cumulative gain (NDCG@k). The predicted top 10% captures, on average, 99.92%, 99.98%, and 99.97% of hourly accesses for the 200-, 500-, and 1,000-blob configurations, respectively. Mean NDCG@k is 0.9984, 0.9987, and 0.9987, respectively. These results indicate that the ranking quality relevant to tier-placement decisions remains stable as the evaluated blob population increases, despite the greater sparsity of the larger configurations. Detailed scalability results are reported in the Experimental Evaluation section.

3. Dynamic hotness ranking

For each decision interval, blobs are ranked according to their predicted next-hour access intensity $\hat{a}_i^{t+1}$. Rather than using a fixed global

access-count threshold, the framework dynamically identifies the top- k predicted blobs as hot candidates:

$$\mathcal{H}_{t+1} = \text{TopK}(\{\hat{a}_i^{t+1}\}_{i=1}^n, k).$$

The number of hot candidates is defined as $k = 0.10n$, such that the predicted top 10% of the evaluated blob population constitutes the desired SSD set at each hourly decision interval. In the primary experimental configuration, $n = 200$ and $k = 20$. For the larger-scale experiments, the same relative-capacity rule is retained, resulting in $k = 50$ for $n = 500$ and $k = 100$ for $n = 1000$. Using the same 10% rule across workload scales enables consistent evaluation of the ranking-based placement mechanism as additional long-tail blobs are included.

This relative ranking is used because cloud access traces are highly skewed and the absolute level of activity can vary substantially across time. The resulting hot set $\mathcal{H}_{t+1}$ represents the desired SSD placement for the next decision interval. This ranking-based formulation replaces the previously considered weighted hotness equation and therefore does not require a manually selected weighting parameter α .

4. Time-varying carbon-aware energy model

To incorporate the temporal environmental impact of electricity consumption into storage-management decisions, the framework uses time-varying grid carbon-intensity signals. Let $CI_{\text{forecast}}(t)$ denote the forecast carbon intensity available to the migration controller during interval t , and let $CI_{\text{actual}}(t)$ denote the corresponding actual carbon intensity used to evaluate realized operational emissions. For an operation consuming $E_i(t)$ joules during interval t , the associated realized operational carbon emissions are calculated as

$$C_i(t) = \frac{E_i(t)}{3.6 \times 10^6} CI_{\text{actual}}(t),$$

where 3.6×10^6 converts joules to kWh. Unlike a constant-carbon-intensity model, this formulation makes the emissions associated with an operation dependent on the time at which the operation is executed.

The historical carbon-intensity trace is aggregated to the same one-hour resolution as the workload. The forecast signal $CI_{\text{forecast}}(t)$ is used exclusively for migration decisions, whereas $CI_{\text{actual}}(t)$ is used only after the decision to calculate realized

emissions. This separation prevents the controller from using future information during migration decisions.

The default low-carbon threshold is derived exclusively from the training portion of the carbon trace rather than from the validation or test periods. Specifically, the median (50th percentile) of the training carbon intensity is used:

$$CI_{\text{th}} = \text{median}(CI_{\text{train}}) = 232 \text{ gCO}_2\text{e/kWh}.$$

A decision interval is considered eligible for migration when

$$CI_{\text{forecast}}(t) \leq CI_{\text{th}}.$$

During forecast high-carbon intervals, placement changes are deferred and the previous SSD/HDD placement is retained. Actual hourly carbon intensity $CI_{\text{actual}}(t)$ is not used to determine whether a migration occurs; it is used only for retrospective emissions accounting.

5. Migration-cost-aware placement decision

At each decision interval, the desired SSD placement obtained from the LSTM hotness ranking is compared with the placement from the preceding interval. A blob entering the predicted top- k set is a candidate for HDD-to-SSD promotion, whereas a blob leaving the set is a candidate for SSD-to-HDD demotion. The energy required for each migration depends on blob size and the modeled read, transfer, and write costs of the source and destination devices.

The placement capacity is expressed through the same relative top- k rule used in the hotness-ranking stage, with $k = 0.10n$. Thus, the primary 200-blob configuration permits 20 blobs in the desired SSD set, while the 500- and 1,000-blob scalability configurations use 50 and 100 blobs, respectively. This keeps the relative SSD allocation constant across workload scales.

For the LSTM Energy-Aware policy, the desired placement update is performed without considering the carbon signal. For the proposed LSTM Carbon-Aware policy, the same placement update is permitted only when the forecast carbon intensity satisfies $CI_{\text{forecast}}(t) \leq CI_{\text{th}}$. Otherwise, the migration is deferred and the previous tier assignment is maintained:

$$\mathbf{P}(t+1) = \begin{cases} \mathbf{P}^*(t+1), & CI_{\text{forecast}}(t) \leq CI_{\text{th}}, \\ \mathbf{P}(t), & CI_{\text{forecast}}(t) > CI_{\text{th}}, \end{cases}$$

where $\mathbf{P}^*(t+1)$ denotes the desired SSD/HDD placement generated from the predicted hot set $\mathcal{H}_{t+1}$, and $\mathbf{P}(t)$ denotes the existing placement at decision interval t . This design allows the Energy-Aware and Carbon-Aware policies to use identical workload predictions, relative SSD allocation, and device-energy assumptions, thereby isolating the effect of incorporating the temporal carbon signal.

The resulting decisions determine the final storage placement of each blob in the tiered storage system consisting of SSD and HDD devices.

3.2. Hotness Prediction Model

Efficient data placement in tiered storage systems requires identifying objects that are likely to be accessed frequently in the near future. To achieve this, we employ an LSTM model that forecasts the next-hour access intensity of each blob using its preceding 24 hours of access history. As illustrated in Figure 1, the predicted access intensities are subsequently used to construct a dynamic ranking, from which the highest-ranked blobs are selected as hot candidates for the next decision interval.

Access Sequence Representation

For each blob, historical access events are aggregated over fixed time windows to construct an access time series. Let $A_i = \{a_i^1, a_i^2, \dots, a_i^t\}$ denote the access sequence for blob i , where a_i^t represents the number of accesses observed during time window t . This representation captures the temporal behavior of object accesses and enables the prediction model to learn workload dynamics.

In the revised implementation, access events are aggregated into one-hour intervals. For each blob, the previous 24 hourly access counts are used as the input sequence for predicting the access intensity in the subsequent hour. Because the access-count distribution is highly skewed, the hourly counts are transformed using $\log(1 + a_i^t)$ before training.

The resulting time-series sequences are used as input to the LSTM model, which predicts the expected number of accesses in the next time window.

LSTM-Based Access Prediction

We employ a Long Short-Term Memory (LSTM) neural network to model temporal dependencies in the access patterns. LSTM networks are well-suited for time-series prediction tasks because they can capture long-term dependencies in sequential data while mitigating the vanishing gradient problem.

Given a 24-hour historical access window for blob i , the LSTM predicts the expected access intensity for the immediately following hour:

$$\hat{a}_i^{t+1} = f_{\text{LSTM}}(a_i^{t-L+1}, \dots, a_i^t), \quad L = 24 \quad (1)$$

where $\hat{a}_i^{t+1}$ denotes the predicted future access frequency for blob i . The prediction output reflects the anticipated workload behavior and is used to estimate the relative importance of each object.

The predicted values are transformed back to the original access-count scale before storage-placement decisions and prediction-performance evaluation are performed.

Dynamic Hotness Ranking

Rather than computing a single static hotness score from a weighted combination of historical and predicted accesses, the revised framework determines the hot set dynamically at each decision interval using the LSTM-predicted next-hour access intensities.

In the original formulation, historical and predicted access frequencies were combined using a weighting parameter α . This formulation required an additional manually selected parameter whose value could affect the resulting hotness scores. To eliminate this dependency, the revised framework removes α entirely and directly ranks objects according to their predicted next-hour access intensity. Consequently, no α value is selected or tuned in the revised model, and sensitivity analysis with respect to α is no longer applicable.

For decision interval $t+1$, all blobs are ranked according to $\hat{a}_i^{t+1}$, and the k blobs with the largest predicted access intensities are classified as hot:

$$\mathcal{H}_{t+1} = \text{TopK}(\{\hat{a}_i^{t+1}\}_{i=1}^n, k) \quad (2)$$

The hot-set size is defined using a fixed relative SSD allocation, $k = 0.10n$. In the primary 200-blob configuration, this results in $k = 20$. For the larger-scale experiments, the same rule yields $k = 50$ for 500 blobs and $k = 100$ for 1,000 blobs. This preserves the same relative SSD allocation across workload scales and allows the ranking mechanism to be evaluated consistently as progressively more long-tail objects are included.

where $\mathcal{H}_{t+1}$ denotes the predicted hot set for the next hourly interval.

The use of a relative top- k ranking avoids introducing an arbitrary absolute hotness threshold. The 10% allocation is used as a controlled experimental capacity rule rather than being tuned to maximize predictive or migration performance.

This dynamic ranking allows a blob to enter or leave the hot set as its predicted workload changes over time. Consequently, storage placement is driven by anticipated near-term demand rather than by a fixed global access threshold or a static popularity score.

3.3. Carbon-Aware Migration Policy

After predicting the next-hour access intensity $\hat{a}_i^{t+1}$ of each blob and identifying the dynamic hot set $\mathcal{H}_{t+1}$, the system constructs the desired SSD/HDD placement $\mathbf{P}^*(t+1)$ and determines whether this placement update should be executed. The objective of the migration policy is to prioritize blobs with high predicted demand while limiting unnecessary migration energy and avoiding migration activity during carbon-intensive grid periods.

The hot-set size follows the relative SSD-allocation rule $k = 0.10n$, so the migration policy operates on the highest-ranked 10% of the evaluated blob population rather than on a fixed number of objects. The primary 200-blob experiment therefore uses $k = 20$, while the larger-scale 500- and 1,000-blob configurations use $k = 50$ and $k = 100$, respectively.

To isolate the effect of temporal carbon awareness, we evaluate two otherwise equivalent predictive policies. The *LSTM Energy-Aware* policy applies the desired placement $\mathbf{P}^*(t+1)$ without considering grid carbon intensity, whereas the proposed *LSTM Carbon-Aware* policy uses forecast carbon intensity to determine whether the same placement update should be executed or deferred.

Accordingly, migration decisions depend on three primary factors: (i) predicted next-hour access intensity $\hat{a}_i^{t+1}$ and dynamic hot-set membership, (ii) size-dependent migration energy, and (iii) forecast time-varying grid carbon intensity. Actual carbon intensity is not used to make migration decisions and is reserved for retrospective emissions evaluation.

Time-Varying Carbon-Aware Constraint

The proposed framework incorporates time-varying grid carbon intensity directly into the migration decision process. Let $CI_{\text{forecast}}(t)$ denote the forecast carbon intensity available to the controller at decision interval t , and let $CI_{\text{actual}}(t)$ denote the corresponding observed carbon intensity used for retrospective emissions evaluation. Both signals are aligned with the workload at an hourly temporal resolution and are expressed in gCO₂e/kWh.

To avoid information leakage, the carbon-intensity threshold is derived exclusively from the training portion of the carbon trace. The default threshold is defined as the median training carbon intensity:

$$CI_{\text{th}} = P_{50}(CI_{\text{train}}) = 232 \text{ gCO}_2\text{e/kWh}. \quad (3)$$

A placement update is eligible for execution only when the forecast carbon intensity does not exceed this threshold:

$$CI_{\text{forecast}}(t) \leq CI_{\text{th}}. \quad (4)$$

If the forecast carbon intensity exceeds the threshold, the proposed Carbon-Aware policy defers the placement update and retains the storage assignment from the previous interval. Importantly, $CI_{\text{actual}}(t)$ is not used to make migration decisions. It is used only to calculate the realized operational carbon emissions after the decisions have been made.

For an operation i consuming $E_i(t)$ joules during interval t , its realized operational carbon emissions are calculated as

$$C_i(t) = \frac{E_i(t)}{3.6 \times 10^6} CI_{\text{actual}}(t), \quad (5)$$

where the factor 3.6×10^6 converts joules to kWh. This formulation allows identical amounts of energy consumed at different times to produce different carbon emissions depending on the temporal carbon intensity of the electricity grid.

Migration Cost Model

Migration overhead is modeled as a size-dependent combination of source-device reads, network transfer, and destination-device writes. Let S_i denote the size of blob i in MB. The time required to read the blob from source tier s , transfer it across the network, and write it to destination tier d is modeled as

$$T_{\text{read}}(i, s) = \frac{S_i}{B_{\text{read},s}}, \quad (6)$$

$$T_{\text{net}}(i) = \frac{S_i}{B_{\text{net}}}, \quad (7)$$

$$T_{\text{write}}(i, d) = \frac{S_i}{B_{\text{write},d}}, \quad (8)$$

where $B_{\text{read},s}$ and $B_{\text{write},d}$ denote the read and write bandwidths of the corresponding storage tiers, and B_{net} denotes the network-transfer bandwidth.

The energy required for a migration from tier s to tier d is then estimated as

$$E_{\text{mig}}(i, s, d) = P_{\text{read},s} T_{\text{read}}(i, s) + P_{\text{net}} T_{\text{net}}(i) + P_{\text{write},d} T_{\text{write}}(i, d) \quad (9)$$

where $P_{\text{read},s}$, $P_{\text{write},d}$, and P_{net} are the corresponding device and transfer power values. Because SSD and HDD devices have different bandwidth and power characteristics, HDD-to-SSD promotion and SSD-to-HDD demotion can incur different migration-energy costs.

The realized carbon emissions associated with a migration executed during interval t are calculated using the actual time-varying carbon intensity:

$$C_{\text{mig}}(i, t) = \frac{E_{\text{mig}}(i, s, d)}{3.6 \times 10^6} CI_{\text{actual}}(t). \quad (10)$$

Table 2. Storage and network parameters used in the analytical evaluation

Parameter	SSD	HDD
Read bandwidth (MB/s)	500	120
Write bandwidth (MB/s)	400	100
Read power (W)	3	6
Write power (W)	4	7
Access latency (ms)	0.2	5.0

Storage and Energy Model Parameters

The storage-device parameters used in the evaluation are analytical modeling parameters rather than measurements from a specific physical deployment. Their relative magnitudes were selected to represent the well-established performance and energy differences between SSD and HDD storage reported in prior storage-system studies [12, 13]. In particular, SSDs provide substantially lower access latency than rotational HDDs, while the power, throughput, and operation duration of each device influence the energy required to service I/O requests. The values used in this study therefore represent a controlled SSD–HDD configuration rather than specifications of a particular commercial device. The same parameters are applied consistently to all evaluated policies.

For migration, the model additionally assumes a network-transfer bandwidth of 125 MB/s and a transfer power of 2 W. Migration energy is not represented by a fixed per-migration constant. Instead, it is computed from the size of the migrated blob and the modeled source-read, network-transfer, and destination-write operations, as defined in the preceding equations. This decomposition follows the general principle that data-movement energy depends on the energy consumed by the components participating in the transfer, including the source, destination, and communication path [14].

Because storage latency, throughput, and power consumption vary across hardware generations, device models, I/O sizes, and workload characteristics, these parameters are not intended to represent a specific commercial device. Rather, they provide a controlled and consistent analytical environment in which all evaluated placement policies are compared under identical assumptions. Physical measurement and hardware-specific calibration of these parameters are left for future deployment-based evaluation.

Migration Decision Rule

At each hourly decision interval t , the LSTM predictions are used to rank the evaluated blobs according to their predicted next-hour access intensities $\hat{a}_i^{t+1}$. The highest-ranked $k = 0.10n$ blobs form the desired hot set $\mathcal{H}_{t+1}$, which defines the target SSD placement $\mathbf{P}^*(t+1)$. Thus,

the migration rule is expressed in relative terms and is not intrinsically tied to the 200-blob configuration used in the primary experiment.

For the LSTM Energy-Aware baseline, changes between the existing placement $\mathbf{P}(t)$ and the desired placement $\mathbf{P}^*(t+1)$ are executed without using a carbon-intensity constraint. The resulting migration energy is nevertheless explicitly included in the total energy cost.

The proposed LSTM Carbon-Aware policy uses the same LSTM predictions, dynamic top- k hot set, relative SSD allocation, and migration-energy model, but conditions placement updates on the forecast carbon intensity:

$$\mathbf{P}(t+1) = \begin{cases} \mathbf{P}^*(t+1), & CI_{\text{forecast}}(t) \leq CI_{\text{th}}, \\ \mathbf{P}(t), & CI_{\text{forecast}}(t) > CI_{\text{th}}. \end{cases} \quad (11)$$

Thus, when the forecast grid carbon intensity is sufficiently low, the controller is allowed to update the SSD/HDD placement toward the predicted hot set. During forecast high-carbon intervals, the placement update is deferred and the existing tier assignment is retained. This formulation shifts flexible migration activity away from carbon-intensive periods without using future actual carbon information.

The distinction between the Energy-Aware and Carbon-Aware policies also provides an ablation for quantifying the contribution of the temporal carbon signal independently of the common LSTM workload prediction and storage-energy models.

4. Experimental Evaluation

This section evaluates the proposed predictive carbon-aware storage tiering framework using the held-out portion of the Azure Functions Blob Access workload. The evaluation examines prediction performance, scalability, access latency, energy consumption, migration overhead, and operational carbon emissions. In addition to the Always-HDD and Always-SSD reference policies, the proposed approach is compared with stronger storage-management baselines, including Least Recently Used (LRU), Least Frequently Used (LFU), and Adaptive Replacement Cache (ARC), under the same SSD-capacity, workload, energy, and latency assumptions.

4.1. Prediction Performance

Before evaluating the storage policies, we first assess the predictive performance of the LSTM model on the held-out test period. The model achieves an R^2 of 0.9197, an MAE of 7.9725 accesses, and an RMSE of 67.9757 accesses on the original access-count scale.

To provide context for these absolute error values, the actual next-hour access counts in the test set have a mean of 27.09, a standard deviation of 239.85, and a range from 0 to 2779 accesses. The MAE therefore corresponds to 29.43% of the mean observed access count and only 0.29% of the full observed range. The RMSE corresponds to 2.45% of the observed range. Its much larger value relative to the MAE (67.98 versus 7.97 accesses) reflects the strongly skewed and sparse nature of the workload, in which a small number of high-access observations contribute disproportionately to squared prediction error.

Because the storage-placement controller selects objects according to their relative predicted demand rather than requiring exact access-count reconstruction, ranking-based metrics provide a more direct assessment of placement-relevant prediction quality. We therefore complement MAE, RMSE, and R^2 with access-coverage and NDCG-based ranking evaluation. At each hourly interval, the k blobs with the highest actual next-hour access counts form the reference top- k set, while the k blobs with the highest predicted access counts form the predicted top- k set. For the primary 200-blob configuration, $k = 20$.

The predicted Top-20 set captures approximately 99.92% of the actual hourly access volume associated with the reference Top-20 objects and achieves a mean NDCG@20 of 0.9984. These ranking-based results indicate that, despite the influence of rare high-access observations on the absolute regression errors, the model preserves the ordering and access concentration that are directly relevant to SSD-placement decisions.

For completeness, the overlap between the predicted and reference Top-20 sets is also evaluated using precision, recall, and F1-score, all of which are 0.7873. These classification-style overlap metrics are reported as complementary measures rather than as the primary criterion for evaluating placement quality, because the workload is highly sparse and the storage controller is fundamentally driven by relative ranking and access concentration.

4.2. Larger-Scale Workload and Ranking Evaluation

To address the limited size of the primary 200-blob configuration, we additionally evaluate the workload representation and prediction pipeline using the 500 and 1,000 most frequently accessed blobs. The same hourly aggregation, 24-hour lookback, chronological data partitioning, LSTM architecture, and relative top- k SSD-selection rule, with $k = 0.10n$, are retained at each scale.

As shown in Table 3, increasing the number of evaluated blobs from 200 to 500 and 1,000 provides only a modest increase in full-trace coverage, from 65.02% to 65.10% and 65.21%, respectively. In contrast,

blob-hour sparsity increases from 93.41% to 96.76% and 97.92%. This behavior confirms that most objects added beyond the primary 200-blob subset belong to a very low-activity long tail.

Table 3. Workload coverage and sparsity at increasing experimental scales

Blobs	Events	Coverage(%)	Nonzero Hours	Sparsity (%)
200	1,300,370	65.02	4,431	93.41
500	1,302,074	65.10	5,443	96.76
1000	1,304,233	65.21	6,980	97.92

The ranking results remain stable across the three workload scales. The predicted top- k sets, with $k = 0.10n$, capture 99.92%, 99.98%, and 99.97% of actual hourly accesses for the 200-, 500-, and 1,000-blob configurations, respectively. Mean NDCG@ k values are 0.9984, 0.9987, and 0.9987, as summarized in Table 4.

Table 4. Ranking performance at increasing workload scales

Blobs	Top- k	Access Coverage (%)	Mean NDCG@ k
200	20	99.92	0.9984
500	50	99.98	0.9987
1000	100	99.97	0.9987

We also evaluate whether the predicted top- k set retrieves blobs that exhibit at least one actual access in the subsequent hour. Active-object recall remains high at 94.74%, 97.00%, and 96.14% for 200, 500, and 1,000 blobs, respectively. However, precision decreases as the workload grows because the number of selected top- k candidates increases from 20 to 100 while the mean number of actually active blobs remains approximately 13–14 per hour. This observation indicates that a fixed percentage-based SSD allocation becomes increasingly over-provisioned as additional low-activity long-tail objects are introduced.

Table 5. Active-object retrieval performance at increasing workload scales

Blobs	Top- k	Active/Hour	Precision	Recall	F1
200	20	12.57	0.5922	0.9474	0.7225
500	50	13.18	0.2518	0.9700	0.3966
1000	100	14.10	0.1325	0.9614	0.2315

Taken together, the larger-scale experiments show that the prediction ranking continues to capture the dominant workload activity as progressively more long-tail blobs are included, while also revealing a limitation of the fixed top- k allocation rule ($k = 0.10n$). Accordingly, the following latency, energy, migration, and carbon analyses use the primary 200-blob configuration, whereas the 500- and 1,000-blob experiments are used specifically to evaluate the scalability of workload representation and predictive ranking.

4.3. Comparison with Strong Storage Baselines

To provide a more representative comparison than the Always-HDD and Always-SSD reference policies alone, we additionally evaluate three widely used storage-management and cache-replacement strategies: Least Recently Used (LRU), Least Frequently Used (LFU), and Adaptive Replacement Cache (ARC). All policies are evaluated using the same 200-blob workload, 51-hour held-out test period, SSD capacity of $k = 20$ blobs, access-energy model, migration-energy model, carbon-intensity trace, and latency assumptions used for the proposed method.

LRU maintains in SSD the objects that have been accessed most recently, whereas LFU prioritizes objects according to their historical access frequency. ARC adaptively balances recency and frequency by maintaining recency- and frequency-oriented replacement lists. These baselines do not use LSTM workload predictions or forecast grid carbon intensity. Consequently, they provide stronger non-predictive reference points for determining whether the proposed predictive carbon-aware policy offers benefits beyond conventional storage-placement strategies.

Because the workload is strongly skewed toward a small number of highly active blobs, LFU represents a particularly competitive baseline. Therefore, the proposed method is not expected to dominate every baseline on every individual metric. Instead, the comparison examines the trade-off among access latency, total energy consumption, migration activity, and operational carbon emissions.

4.4. Latency Analysis

Figure 2 compares the average access latency across the reference, conventional replacement, and predictive storage policies over the held-out test period for the primary 200-blob configuration

As expected, Always-HDD exhibits the highest average latency at 5.000 ms per operation, while Always-SSD provides the theoretical lower reference at 0.200 ms per operation. Among the stronger baselines, LRU, LFU, and ARC achieve average latencies of 0.2091, 0.2043, and 0.2015 ms per operation, respectively. The LSTM Energy-Aware and LSTM Carbon-Aware policies achieve 0.2037 and 0.2036 ms per operation, respectively.

The proposed Carbon-Aware policy therefore achieves lower average latency than both LRU and LFU, although ARC provides slightly lower latency. Importantly, incorporating the carbon-aware migration constraint produces essentially no latency penalty relative to the otherwise equivalent LSTM Energy-Aware policy. These results indicate that the proposed policy can reduce migration activity and carbon emissions while maintaining access performance

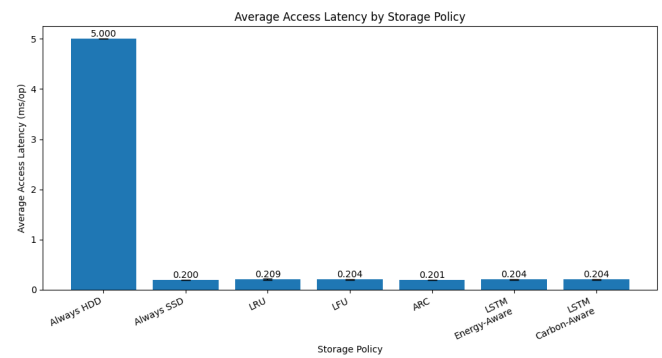


Figure 2. Average access latency across reference, conventional replacement, and predictive storage policies during the held-out test period. Error bars indicate 95% confidence intervals estimated using a six-hour moving-block bootstrap over the 51 hourly test intervals.

close to conventional high-performance placement strategies.

4.5. Energy and Migration Analysis

Figure 3 compares the total energy consumption across the seven evaluated storage policies for the primary 200-blob configuration. In addition to the Always-HDD, Always-SSD, and two LSTM-based policies, the comparison includes LRU, LFU, and ARC as stronger conventional storage-management baselines. Access energy and migration energy are both included in the reported total energy

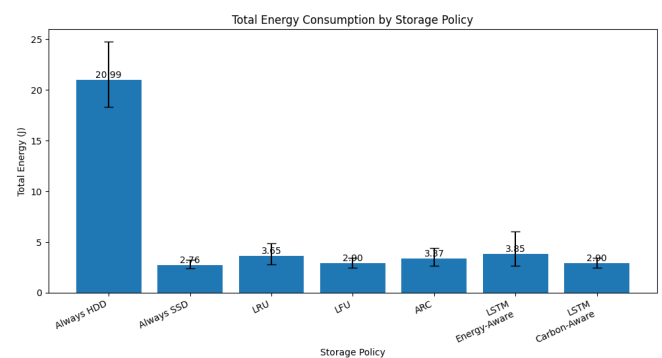


Figure 3. Total energy consumption across storage policies during the held-out test period. Error bars indicate 95% confidence intervals estimated using a six-hour moving-block bootstrap over the 51 hourly test intervals.

Always-HDD consumes 20.9939 J, reflecting the higher access-energy cost of the HDD tier, whereas Always-SSD consumes 2.7586 J. Among the conventional replacement policies, LRU consumes 3.6534 J, LFU consumes 2.8973 J, and ARC consumes 3.3732 J. The LSTM Energy-Aware policy consumes 3.8507 J, including 0.9545 J of migration energy. In contrast, the

Table 6. Energy, migration, carbon, and latency results across storage policies

Policy	Energy (J)	Mig. Energy (J)	Migrations	Carbon (gCO ₂ e)	Latency (ms/op)
Always HDD	20.9939	0.0000	-	0.001339	5.0000
Always SSD	2.7586	0.0000	-	0.000176	0.2000
LRU	3.6534	0.5203	68	0.000234	0.2091
LFU	2.8973	0.0000	0	0.000186	0.2043
ARC	3.3732	0.4829	46	0.000217	0.2015
LSTM Energy-Aware	3.8507	0.9545	84	0.000257	0.2037
LSTM Carbon-Aware	2.8995	0.0043	42	0.000186	0.2036

proposed LSTM Carbon-Aware policy consumes 2.8995 J, including only 0.0043 J of migration energy.

Relative to the LSTM Energy-Aware baseline, the Carbon-Aware policy reduces total energy consumption by 24.70%. This reduction is primarily associated with fewer placement changes during the test period. Compared with the stronger conventional baselines, the Carbon-Aware policy also consumes 20.63% less total energy than LRU and 14.04% less than ARC, while its total energy consumption is essentially comparable to LFU.

Figure 4 compares the number of migration operations among the capacity-constrained storage-placement policies. LRU performs 68 migrations, ARC performs 46 migrations, and the LSTM Energy-Aware policy performs 84 migrations during the 51-hour held-out test period. LFU performs no migrations during the test period because its frequency-based resident set remains unchanged after initialization. The proposed LSTM Carbon-Aware policy performs 42 migrations, representing a 50% reduction relative to the otherwise equivalent LSTM Energy-Aware policy. It also requires fewer placement changes than both LRU and ARC.

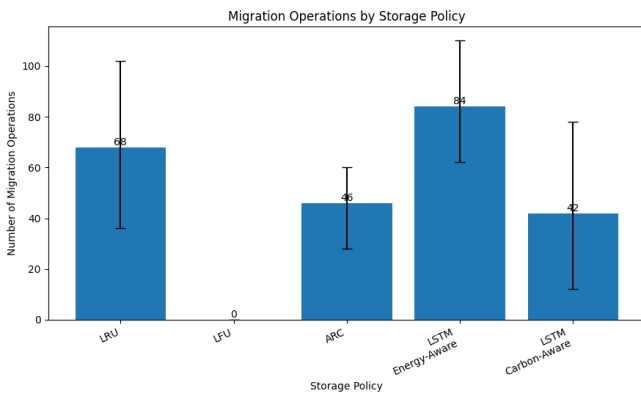


Figure 4. Number of migration operations across conventional replacement and predictive storage policies during the held-out test period. Error bars indicate 95% confidence intervals estimated using a six-hour moving-block bootstrap over the 51 hourly test intervals

The stronger-baseline comparison reveals an important trade-off. LFU achieves slightly lower total energy consumption than the proposed Carbon-Aware policy

(2.8973 J versus 2.8995 J) and performs no migrations during the test period. However, its average access latency is higher at 0.2043 ms/op compared with 0.2036 ms/op for the proposed policy. ARC achieves the lowest latency among the conventional replacement policies at 0.2015 ms/op, but requires 46 migrations and consumes 3.3732 J. LRU performs less favorably across these metrics, with 68 migrations, 3.6534 J of total energy, and an average latency of 0.2091 ms/op. These results show that the proposed method does not uniformly dominate every baseline on every individual metric; rather, it provides a competitive balance among access latency, energy consumption, migration overhead, and carbon emissions.

Figure 5 further compares the operational carbon emissions of all seven policies. Always-HDD produces the highest emissions at 0.001339 gCO₂e, followed by the LSTM Energy-Aware policy at 0.000257 gCO₂e. LRU and ARC produce 0.000234 and 0.000217 gCO₂e, respectively. The proposed LSTM Carbon-Aware policy produces approximately 0.000186 gCO₂e, which is substantially lower than LRU, ARC, and the LSTM Energy-Aware baseline and approximately equal to LFU under the reported precision.

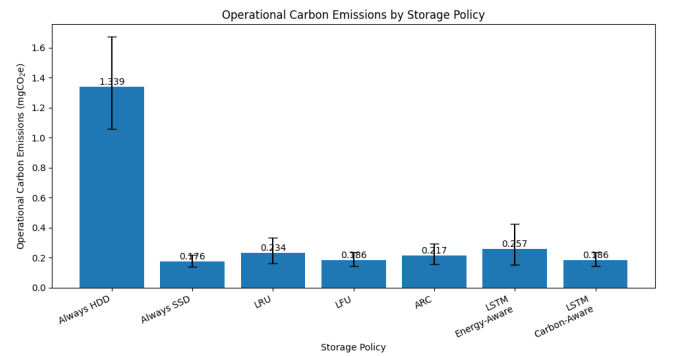


Figure 5. Operational carbon emissions across storage policies during the held-out test period. Error bars indicate 95% confidence intervals estimated using a six-hour moving-block bootstrap over the 51 hourly test intervals

4.6. Carbon-Aware Evaluation

To directly evaluate the benefit of time-varying carbon-aware scheduling, the LSTM Carbon-Aware policy is compared with the otherwise equivalent LSTM Energy-Aware baseline under the primary 200-blob configuration. Both policies use the same workload predictions, dynamic top- k hot set $\mathcal{H}_t$, relative SSD allocation $k = 0.10n$, storage-energy model, and migration-energy model. For this primary configuration, $n = 200$ and $k = 20$. The only difference is whether $CI_{\text{forecast}}(t)$ is used to defer placement changes.

Over the held-out test period, the LSTM Energy-Aware policy produces 0.000257106 gCO₂e, whereas

the LSTM Carbon-Aware policy produces 0.000185745 gCO₂e. This corresponds to a 27.76% reduction in operational carbon emissions.

To distinguish the contribution of reduced energy consumption from the additional benefit of temporal carbon-aware scheduling, we evaluate a constant-carbon-intensity counterfactual using the mean test-period actual carbon intensity. Under this constant-CI assumption, the Carbon-Aware policy achieves a 24.70% reduction in emissions, equal to its reduction in total energy consumption. When the time-varying $CI_{\text{actual}}(t)$ signal is used for emissions accounting, the reduction increases to 27.76%. Therefore, temporal carbon awareness contributes an additional 3.05 percentage points of emissions reduction beyond the effect of lower energy consumption alone.

The same effect is observed in the carbon intensity associated specifically with migration activity. The migration-energy-weighted carbon intensity decreases from 269.66 gCO₂e/kWh for the Energy-Aware policy to 174.65 gCO₂e/kWh for the Carbon-Aware policy, corresponding to a 35.23% reduction in the carbon intensity of electricity consumed during migration.

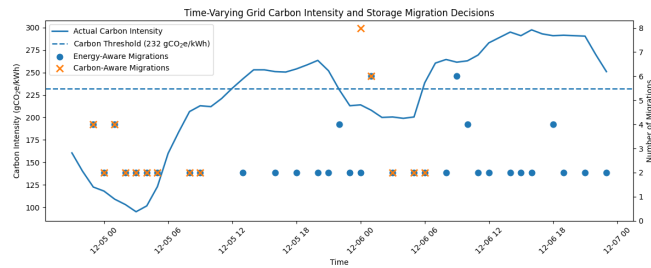


Figure 6. Time-varying grid carbon intensity and migration decisions during the held-out test period

Figure 6 illustrates the temporal relationship between $CI_{\text{actual}}(t)$ and migration activity. The Energy-Aware policy continues to perform migrations during several high-carbon periods, whereas the Carbon-Aware policy suppresses or defers many of these placement changes. Because migration decisions are based on $CI_{\text{forecast}}(t)$ while the figure shows $CI_{\text{actual}}(t)$, occasional migration events may occur when realized carbon intensity is above $CI_{\text{threshold}}$; this reflects forecast error rather than the use of future information.

4.7. Carbon-Threshold Sensitivity

To evaluate whether the observed carbon benefit in the primary 200-blob configuration depends on a narrowly tuned carbon threshold, we repeat the Carbon-Aware simulation using thresholds corresponding to the 25th, 40th, 50th, 60th, and 75th percentiles of the training carbon-intensity distribution.

Table 7. Sensitivity of the Carbon-Aware policy to the training carbon-intensity threshold

Percentile	CI_{th}	Allowed Hours	Migrations	Carbon Red. (%)	Latency
25	185.5	13	22	27.79	0.2038
40	210.6	19	36	27.77	0.2036
50	232.0	24	42	27.76	0.2036
60	243.6	27	44	27.72	0.2037
75	262.0	32	56	27.66	0.2039

Across the tested threshold range, the carbon reduction remains between 27.66% and 27.79%, while average latency remains between 0.2036 and 0.2039 ms per operation. Although the number of migrations increases as the threshold becomes less restrictive, the overall carbon reduction remains nearly unchanged. These results indicate that the observed carbon benefit is robust to the precise threshold choice.

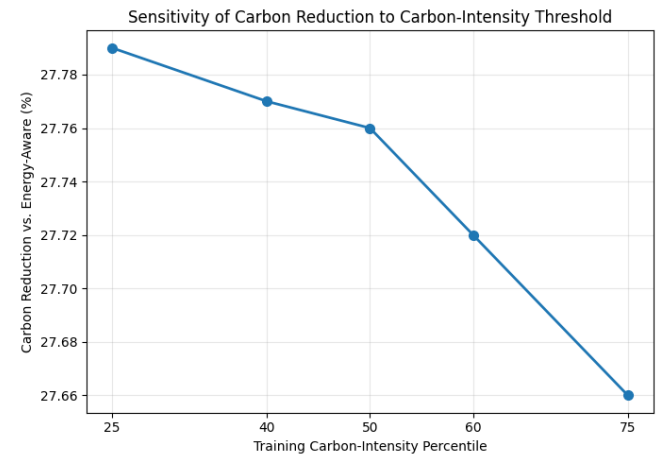


Figure 7. Carbon-emission reduction as a function of the training carbon-intensity threshold percentile

Overall, the experimental results from the primary 200-blob configuration demonstrate that incorporating the forecast carbon-intensity signal $CI_{\text{forecast}}(t)$ into predictive storage-tiering decisions reduces both migration activity and operational carbon emissions while preserving access latency. The larger-scale 500- and 1,000-blob experiments presented earlier are used to evaluate workload and prediction scalability rather than to claim that the full carbon-aware policy was simulated at those larger scales. The ablation analysis further confirms that the observed improvement in the primary configuration is not solely an artifact of reduced energy consumption, as temporal carbon-aware scheduling provides an additional measurable emissions benefit.

4.8. Statistical Significance Analysis

To quantify the variability and statistical significance of the experimental results, we analyze the performance of each capacity-constrained storage policy over the 51 hourly intervals of the held-out test period. For each

metric, we report the hourly mean, standard deviation (SD), and 95% confidence interval. Because consecutive workload intervals may exhibit temporal dependence, the confidence intervals are estimated using a moving-block bootstrap with a six-hour block length.

Table 8 summarizes the hourly descriptive statistics for LRU, LFU, ARC, LSTM Energy-Aware, and the proposed LSTM Carbon-Aware policy. The results show that the Carbon-Aware policy achieves a mean hourly energy consumption of 0.056853 ± 0.040747 J/hour, with a 95% confidence interval of $[0.048809, 0.068071]$ J/hour. Its mean hourly operational carbon emissions are approximately 4×10^{-6} gCO₂e/hour, with a 95% confidence interval of approximately $[3 \times 10^{-6}, 5 \times 10^{-6}]$ gCO₂e/hour. The mean access latency is 0.203723 ± 0.015743 ms/op, with a 95% confidence interval of $[0.200599, 0.206708]$ ms/op, and the mean migration frequency is 0.823529 ± 1.657780 migrations/hour, with a 95% confidence interval of $[0.235294, 1.529412]$.

Statistical significance is further evaluated using paired Wilcoxon signed-rank tests between the proposed Carbon-Aware policy and each competing capacity-constrained policy. The tests are performed on paired hourly observations. To account for multiple comparisons, the resulting p -values are adjusted using the Holm procedure, with statistical significance assessed at a significance level of 0.05.

The statistical tests provide particularly strong support for the comparison with the LSTM Energy-Aware baseline, which isolates the effect of incorporating the temporal carbon signal while keeping the prediction model and storage assumptions unchanged. The proposed Carbon-Aware policy achieves statistically significant reductions in hourly total energy consumption ($p_{\text{Holm}} = 0.000287$), operational carbon emissions ($p_{\text{Holm}} = 0.000287$), and migration frequency ($p_{\text{Holm}} = 0.000375$) relative to the LSTM Energy-Aware policy. In contrast, the latency difference is not statistically significant ($p_{\text{Holm}} = 0.628727$), providing additional evidence that carbon-aware migration deferral does not introduce a measurable access-latency penalty. The comparisons with the conventional caching baselines provide a more nuanced result. Relative to LRU, none of the four hourly metrics shows a statistically significant difference after Holm correction. Relative to LFU, the differences in energy and carbon emissions are not statistically significant; however, the Carbon-Aware policy achieves significantly lower access latency ($p_{\text{Holm}} = 0.001151$). LFU performs significantly fewer migrations ($p_{\text{Holm}} = 0.000570$), because its resident set remains unchanged during the evaluated test period.

Relative to ARC, the differences in hourly energy consumption, carbon emissions, and migration frequency are not statistically significant. ARC achieves a statistically lower access latency ($p_{\text{Holm}} = 0.008758$), although

the absolute difference is small: the mean hourly latency is 0.201483 ms/op for ARC and 0.203723 ms/op for the proposed Carbon-Aware policy. Overall, these results show that the proposed policy does not statistically dominate every baseline on every metric. Rather, its main statistically supported advantage is the simultaneous reduction of energy consumption, operational carbon emissions, and migration activity relative to the otherwise equivalent predictive Energy-Aware policy, without a statistically significant latency penalty.

5. Discussion

The experimental results demonstrate that predictive storage management can significantly improve the efficiency of tiered storage systems while maintaining competitive performance. In this section, we analyze the key insights obtained from the evaluation and discuss the implications of the proposed approach.

5.1. Trade-off Between Migration Cost and Performance

One of the primary challenges in tiered storage systems is balancing migration overhead with performance improvements. The Always-SSD policy provides the lowest access latency because all requests are served from the faster storage tier, whereas the Always-HDD policy avoids migration overhead but incurs substantially higher access latency and energy consumption.

The predictive policies provide a practical middle ground by maintaining only the dynamically selected highest-ranked objects in the SSD tier. In the primary 200-blob configuration, the framework selects the predicted top- k blobs with $k = 20$, corresponding to 10% of the evaluated objects, at each decision interval. Rather than relying solely on a forced binary top- k ground-truth classification, which is problematic in this highly sparse workload because many hourly intervals contain fewer than k active blobs, we also evaluate the ranking using access coverage and NDCG@ k . For the primary configuration ($k = 20$), the predicted top- k set captures 99.92% of actual hourly accesses on average and achieves a mean NDCG@ k of 0.9984.

The Energy-Aware and Carbon-Aware policies achieved average access latencies of 0.2037 and 0.2036 ms per operation, respectively, compared with 0.2 ms for Always-SSD and 5 ms for Always-HDD. Importantly, introducing the carbon-aware constraint did not produce a measurable performance penalty: the Carbon-Aware policy exhibited a -0.07% latency difference relative to the Energy-Aware policy. Thus, the reduction in migration activity and carbon emissions was achieved without degrading workload-level access performance.

The comparison with stronger conventional baselines further clarifies this trade-off. LRU, LFU, and ARC

Table 8. Hourly descriptive statistics over the 51-hour held-out test period. Confidence intervals are estimated using a six-hour moving-block bootstrap

Policy	Metric	Mean	SD	95% Conf. Lower	95% Conf. Upper
LRU	Energy (J/hour)	0.071635	0.070759	0.055364	0.095141
LRU	Carbon (gCO ₂ e/hour)	0.000005	0.000005	0.000003	0.000007
LRU	Latency (ms/op)	0.209044	0.054580	0.200758	0.224415
LRU	Migrations/hour	1.333333	2.487301	0.745098	2.039216
LFU	Energy (J/hour)	0.056809	0.040743	0.048659	0.068109
LFU	Carbon (gCO ₂ e/hour)	0.000004	0.000003	0.000003	0.000005
LFU	Latency (ms/op)	0.204483	0.015485	0.201378	0.207104
LFU	Migrations/hour	0.000000	0.000000	0.000000	0.000000
ARC	Energy (J/hour)	0.066140	0.064644	0.052766	0.086435
ARC	Carbon (gCO ₂ e/hour)	0.000004	0.000004	0.000003	0.000006
ARC	Latency (ms/op)	0.201483	0.004193	0.200630	0.202560
ARC	Migrations/hour	0.901961	1.565310	0.588235	1.176471
LSTM Energy-Aware	Energy (J/hour)	0.075504	0.078654	0.051332	0.117570
LSTM Energy-Aware	Carbon (gCO ₂ e/hour)	0.000005	0.000006	0.000003	0.000008
LSTM Energy-Aware	Latency (ms/op)	0.203855	0.015754	0.200646	0.206878
LSTM Energy-Aware	Migrations/hour	1.647059	1.533930	1.215686	2.156863
LSTM Carbon-Aware	Energy (J/hour)	0.056853	0.040747	0.048809	0.068071
LSTM Carbon-Aware	Carbon (gCO ₂ e/hour)	0.000004	0.000003	0.000003	0.000005
LSTM Carbon-Aware	Latency (ms/op)	0.203723	0.015743	0.200599	0.206708
LSTM Carbon-Aware	Migrations/hour	0.823529	1.657780	0.235294	1.529412

Table 9. Paired Wilcoxon signed-rank tests comparing the proposed LSTM Carbon-Aware policy with competing policies. Reported *p*-values are Holm-adjusted for multiple comparisons

Metric	Comparison	Wilcoxon Statistic	Holm-adjusted <i>p</i>	Significant
Energy	Carbon-Aware vs. LSTM Energy-Aware	74.0	0.000287	Yes
Energy	Carbon-Aware vs. LRU	469.0	0.273174	No
Energy	Carbon-Aware vs. LFU	616.0	1.000000	No
Energy	Carbon-Aware vs. ARC	616.0	1.000000	No
Carbon	Carbon-Aware vs. LSTM Energy-Aware	74.0	0.000287	Yes
Carbon	Carbon-Aware vs. LRU	454.0	0.201439	No
Carbon	Carbon-Aware vs. LFU	603.0	1.000000	No
Carbon	Carbon-Aware vs. ARC	603.0	1.000000	No
Latency	Carbon-Aware vs. LSTM Energy-Aware	96.0	0.628727	No
Latency	Carbon-Aware vs. LRU	445.0	0.406733	No
Latency	Carbon-Aware vs. LFU	228.0	0.001151	Yes
Latency	Carbon-Aware vs. ARC	267.5	0.008758	Yes
Migrations	Carbon-Aware vs. LSTM Energy-Aware	50.5	0.000375	Yes
Migrations	Carbon-Aware vs. LRU	322.5	0.241977	No
Migrations	Carbon-Aware vs. LFU	0.0	0.000570	Yes
Migrations	Carbon-Aware vs. ARC	436.5	0.808136	No

achieve average access latencies of 0.2091, 0.2043, and 0.2015 ms/op, respectively, compared with 0.2036 ms/op for the proposed LSTM Carbon-Aware policy. Thus, the proposed method improves latency relative to LRU and LFU, although ARC achieves slightly lower latency. These results indicate that the benefit of the proposed approach should be interpreted as a multi-objective trade-off rather than uniform dominance on every individual metric.

5.2. Energy and Migration Efficiency

Energy consumption is an important consideration in large-scale cloud storage systems because both data accesses and migrations contribute to operational energy use.

During the 51-hour held-out test period, the Always-HDD baseline consumed 20.9939 J, whereas the Always-SSD baseline consumed 2.7586 J. The large difference results from the device-level access-energy assumptions used in our model, under which HDD

reads and writes require substantially more energy than their SSD counterparts.

The LSTM Energy-Aware policy consumed 3.8507 J in total, consisting of 2.8962 J of access energy and 0.9545 J of migration energy. In comparison, the proposed LSTM Carbon-Aware policy consumed 2.8995 J, including 2.8952 J of access energy and only 0.0043 J of migration energy. This corresponds to a 24.70% reduction in total energy relative to the Energy-Aware policy.

The stronger baseline comparison provides additional context for these results. LRU consumes 3.6534 J and ARC consumes 3.3732 J, meaning that the proposed Carbon-Aware policy reduces total energy by approximately 20.63% and 14.04%, respectively. LFU consumes 2.8973 J, which is essentially identical to the 2.8995 J consumed by the proposed policy. Therefore, the Carbon-Aware policy is competitive with the most energy-efficient conventional baseline while consuming substantially less energy than LRU, ARC, and the LSTM Energy-Aware policy.

Migration behavior further differentiates the policies. LRU performs 68 migrations, ARC performs 46, and the LSTM Energy-Aware policy performs 84 during the test period. The proposed Carbon-Aware policy performs 42 migrations, representing a 50% reduction relative to the Energy-Aware policy and fewer placement changes than both LRU and ARC. LFU performs no migrations during the test period because its frequency-based resident set remains unchanged after initialization. Although LFU therefore has lower migration overhead, its average access latency is higher than that of the proposed policy. Taken together, these results show that the Carbon-Aware approach achieves a competitive balance among energy consumption, migration overhead, and access latency rather than optimizing any one metric in isolation.

5.3. Benefits of Time-Varying Carbon Awareness

A central result of this study is that the benefit of the proposed policy is not explained solely by reduced energy consumption. Unlike a constant-carbon model, the proposed framework uses time-varying grid carbon intensity to determine when migration activity should occur. During the test period, $CI_{\text{actual}}(t)$ ranged from 95.0 to 297.5 gCO₂e/kWh, with a mean of 224.25 gCO₂e/kWh. The carbon-intensity threshold CI_{th} was derived exclusively from the training period as its median value, 232 gCO₂e/kWh.

The Energy-Aware policy produced 0.0002571 gCO₂e during the test period, whereas the Carbon-Aware policy produced 0.0001857 gCO₂e, corresponding to a 27.76% reduction in operational carbon emissions. To distinguish the benefit of temporal carbon awareness from the benefit of simply consuming less energy,

we additionally evaluated a constant-carbon-intensity counterfactual. Under constant carbon intensity, the reduction was 24.70%, whereas the reduction under $CI_{\text{actual}}(t)$ was 27.76%. Therefore, temporal carbon awareness contributed an additional 3.05 percentage points of carbon reduction beyond the energy-saving effect alone.

The stronger baselines also demonstrate that this carbon performance is competitive with established replacement strategies. LRU and ARC produce approximately 0.000234 and 0.000217 gCO₂e, respectively, compared with approximately 0.000186 gCO₂e for the proposed Carbon-Aware policy. LFU also produces approximately 0.000186 gCO₂e at the reported precision. Thus, the proposed policy achieves substantially lower operational emissions than LRU and ARC and approximately matches the low emissions of LFU, while providing lower average access latency than LFU.

This effect is also evident in the timing of migration operations. The migration-energy-weighted carbon intensity decreased from 269.66 gCO₂e/kWh for the Energy-Aware policy to 174.65 gCO₂e/kWh for the Carbon-Aware policy, a 35.23% reduction. This result provides direct evidence that the proposed policy not only reduces migration activity but also shifts the remaining migration energy toward cleaner grid periods.

5.4. Workload Scale and Dataset Selection

The primary experiments focus on the 200 most frequently accessed blobs rather than on a random subset of the complete trace. This selection is motivated by the strongly concentrated workload: the 200 selected blobs represent only 0.0435% of the 459,580 unique blobs in the processed trace but account for 65.02% of all 2,000,000 access events. Thus, the primary configuration captures the objects responsible for most storage activity while avoiding the computational cost of modeling hundreds of thousands of predominantly low-activity objects.

However, this subset should not be interpreted as preserving every characteristic of the full workload. In particular, the complete trace contains 74.10% reads and 25.90% writes, whereas the top-200 subset contains 98.84% reads and 1.16% writes. The selected workload therefore preserves the dominant access concentration and heavy-tailed behavior relevant to hot-data tiering, but it is substantially more read-dominated than the full trace.

To evaluate the effect of increasing workload scale, we also constructed 500- and 1,000-blob configurations. Increasing the evaluated population from 200 to 500 and 1,000 blobs increased full-trace event coverage only from 65.02% to 65.10% and 65.21%, respectively, while blob-hour sparsity increased from 93.41% to 96.76%

and 97.92%. These results indicate that most of the additional objects belong to the low-activity long tail of the workload.

The ranking behavior also remains stable with respect to access concentration. Using the relative hot-set definition $k = 0.10n$, the predicted top- k sets capture 99.92%, 99.98%, and 99.97% of actual hourly accesses for the $n = 200$, $n = 500$, and $n = 1000$ configurations, respectively, while mean NDCG@ k remains between 0.9984 and 0.9987. Active-object recall also remains high at 94.74%, 97.00%, and 96.14%, respectively. However, active-object precision decreases from 59.22% to 25.18% and 13.25% as the workload scale increases. This occurs because the relative allocation $k = 0.10n$ increases k from 20 to 50 and 100, respectively, while the mean number of actually active blobs remains approximately 13–14 per hour. The scalability experiment therefore supports the stability of the ranking for dominant accesses while also revealing the limitations of a fixed percentage-based SSD allocation in increasingly sparse workloads.

5.5. Sensitivity to the Carbon-Intensity Threshold

The sensitivity analysis further shows that the framework is relatively stable across different carbon-intensity thresholds. Training percentiles between the 25th and 75th percentiles produced carbon reductions ranging only from 27.79% to 27.66% relative to the Energy-Aware policy. However, the number of migrations increased from 22 at the 25th-percentile threshold to 56 at the 75th-percentile threshold as more test hours became eligible for migration.

We use the 50th percentile of the training carbon intensity, corresponding to $CI_{th} = 232$ gCO₂e/kWh, as the primary operating point. This choice is not intended to maximize carbon reduction—the 25th percentile yields a marginally larger reduction—but instead provides a moderate operating point between restricting migration opportunities and allowing more frequent tier transitions. At this threshold, the policy performs 42 migrations and achieves a 27.76% carbon reduction.

5.6. Limitations and Future Directions

Several limitations should be considered when interpreting these results. First, the evaluation is trace-driven rather than deployment-based. Although the access traces and carbon-intensity signal are time-varying, device energy and latency are estimated analytically from assumed SSD, HDD, and network bandwidth and power parameters. Consequently, the reported absolute energy and carbon values should be interpreted as model-based estimates, while the relative policy comparisons are the primary focus.

Second, although the workload and prediction analysis was extended from 200 to 500 and 1,000 blobs, the full energy, migration, latency, and carbon-aware policy evaluation was conducted on the primary 200-blob configuration. The larger-scale experiments were designed specifically to examine workload coverage, sparsity, and predictive ranking behavior. Future work should extend the complete carbon-aware migration simulation to substantially larger object populations and additional cloud traces.

Third, the activity-focused top-200 subset does not preserve the read/write composition of the complete trace. Because the selected objects are dominated by reads, the reported policy results may not generalize directly to more write-intensive workloads. Evaluating the framework on workloads with different read/write ratios will therefore be important for assessing broader applicability.

Fourth, the current system considers a two-tier SSD–HDD architecture and uses a fixed relative top- k SSD-placement rule with $k = 0.10n$. The larger-scale experiments show that this rule becomes increasingly over-provisioned as low-activity long-tail objects are added. Future work could incorporate adaptive hot-set sizing, explicit byte-level SSD capacity constraints, heterogeneous storage technologies, dynamic capacity allocation, and migration-rate limits.

Finally, although the LSTM achieves an R^2 of 0.9197 for access-count prediction and the predicted top- k set ($k = 20$ in the primary configuration) captures 99.92% of hourly accesses, the increasing sparsity of larger workloads highlights the need for prediction objectives that explicitly account for ranking quality and active-object sparsity. Future work could investigate ranking-oriented loss functions, probabilistic hotness prediction, adaptive retraining, and longer forecasting horizons.

6. Conclusion

In this paper, we investigated predictive and carbon-aware data placement in tiered cloud storage systems. The proposed framework combines workload forecasting with storage-tier placement and time-varying grid carbon intensity to determine both which objects should occupy the SSD tier and when migration operations should be performed.

Using the Azure Functions Blob Access workload, we constructed hourly access sequences for the 200 most frequently accessed blobs over 336 hours and trained an LSTM using a 24-hour lookback window. These 200 blobs account for 65.02% of the 2,000,000 access events in the processed trace. The model achieved an R^2 of 0.9197 and an MAE of 7.9725 accesses on the original access-count scale. In the primary configuration, where $n = 200$ and $k = 20$, the predicted top- k set captured

99.92% of actual hourly accesses on average and achieved a mean NDCG@ k of 0.9984.

Larger-scale experiments using $n = 500$ and $n = 1000$ blobs showed that access coverage of the predicted top- k sets, with $k = 0.10n$, remained above 99.9%, while mean NDCG@ k remained approximately 0.999 despite workload sparsity increasing to 96.76% and 97.92%. These experiments indicate that the predictive ranking continues to capture the dominant access activity as progressively more low-activity objects are included, while also revealing that fixed percentage-based SSD allocation becomes increasingly over-provisioned in highly sparse workloads.

The experimental results demonstrate that incorporating temporal carbon information can substantially improve the efficiency of predictive tiering. Relative to the LSTM Energy-Aware policy, the proposed Carbon-Aware policy reduced operational carbon emissions by 27.76%, total energy consumption by 24.70%, and the number of migrations by 50%, while changing average access latency by only -0.07% .

The ablation analysis further showed that 24.70% of the carbon reduction can be explained by lower energy consumption under a constant-carbon counterfactual, while evaluation using the time-varying actual carbon-intensity signal $CI_{\text{actual}}(t)$ provides an additional 3.05 percentage points of reduction. Moreover, migration-weighted carbon intensity decreased from 269.66 to 174.65 gCO₂e/kWh, demonstrating that the framework schedules migration energy during substantially cleaner grid periods.

Overall, these findings show that workload prediction and temporal carbon signals can be jointly exploited to reduce storage migration overhead and operational emissions without sacrificing access performance in the primary workload configuration. The larger-scale analysis provides additional evidence that the dominant access behavior remains predictable as more long-tail objects are included, while highlighting the need for adaptive SSD allocation and broader full-policy evaluation at larger scales.

References

- [1] H. Han and M. Song, "QoE-aware video storage power management based on hot and cold data classification," in *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2018, pp. 7–12.
- [2] Y. Guan, Z. Ma, and L. Li, "HDFS optimization strategy based on hierarchical storage of hot and cold data," *Procedia CIRP*, vol. 83, pp. 415–418, 2019.
- [3] S. Ma, T. Zhao, J. Gu, and Y. Wang, "A hierarchical storage mechanism for hot and cold data based on temperature model," in *International Conference on Database and Expert Systems Applications*. Springer, 2024, pp. 153–159.
- [4] X. Xie, "MQHD: Dynamic load balancing method for energy-aware storage systems," *Scientific Programming*, vol. 2020, no. 1, Art. no. 6843572, 2020.
- [5] S. Zhizhuo, T. Yu-An, and L. Yuanzhang, "An energy-efficient storage for video surveillance," *Multimedia Tools and Applications*, vol. 73, no. 1, pp. 151–167, 2014.
- [6] M. Darwich, Y. Ismail, T. Darwich, and M. Bayoumi, "Cost-efficient storage for on-demand video streaming on cloud," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE, 2020, pp. 1–4.
- [7] S. Raut, A. Saxena, W. Leggette, S. Patil, V. Hsu, and D. Archer, "Workload management for sustainability with storage systems in hybrid cloud deployment," in *2023 IEEE Women in Technology Conference (WINTeCHCON)*. IEEE, 2023, pp. 1–6.
- [8] S. Köhler, B. Herzog, H. Herzog, M. Vögele, L. Wenzel, A. Polze, and T. Hönig, "Carbon-aware memory placement," *ACM SIGENERGY Energy Informatics Review*, vol. 4, no. 3, pp. 39–45, 2024.
- [9] M. Gebhard, "Carbon-aware systemd timers," *Science*, 2024.
- [10] V. Rao and A. A. Chien, "Understanding the operational carbon footprint of storage reliability and management," *ACM SIGENERGY Energy Informatics Review*, vol. 4, no. 5, pp. 180–187, 2024.
- [11] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing carbon: The elusive environmental footprint of computing," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 854–867.
- [12] Y. Li and D. D. E. Long, "Which storage device is the greenest? Modeling the energy cost of I/O workloads," in *22nd IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2014.
- [13] E. Borba, E. Tavares, and P. Maciel, "A modeling approach for estimating performance and energy consumption of storage systems," *Journal of Computer and System Sciences*, 2022.
- [14] V. De Maio, R. Prodan, S. Benedict, and G. Kecskemeti, "Modelling energy consumption of network transfers and virtual machine migration," *Future Generation Computer Systems*, vol. 56, pp. 388–406, 2016.