2022 IEEE/ACM 8th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD) | 978-1-6654-6337-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/DRBSD56682.2022.00013

Characterization of Transform-Based Lossy Compression for HPC Datasets

Aekyeung Moon
ETRI
akmoon@etri.re.kr

Jiaxi Chen
UMass Lowell
jiaxi_chen@student.uml.edu

Seung Woo Son
UMass Lowell
seungwoo_son@uml.edu

Minjun Kim
Andong National Univ.
manjun1004@gmail.com

***Abstract*—As the scale and complexity of high-performance computing (HPC) systems keep growing, data compression techniques are often adopted to reduce the data volume and processing time. While lossy compression becomes preferable to a lossless one because of the potential benefit of generating a high compression ratio, it would lose its worth the effort without finding an optimal balance between volume reduction and information loss. Among many lossy compression techniques, transform-based lossy algorithms utilize spatial redundancy better. However, the transform-based lossy compressor has received relatively less attention because there is a lack of understanding of its compression performance on scientific data sets. The insight of this paper is that, in transform-based lossy compressors, quantifying dominant coefficients at the block level reveals the right balance, potentially impacting overall compression ratios. Motivated by this, we characterize three transformation-based lossy compression mechanisms with different information compaction methods using the statistical features that capture data characteristics. And then, we build several prediction models using the statistical features and the characteristics of dominant coefficients and evaluate the effectiveness of each model using six HPC datasets from three production-level simulations at scale. Our results demonstrate that the random forest classifier captures the behavior of dominant coefficients precisely, achieving nearly 99% of prediction accuracy.**

***Index Terms*—Lossy Compression, Discrete Cosine Transform (DCT), Prediction Models, Data Fidelity**

I. Introduction

Data compression is increasingly becoming crucial in various HPC application domains, especially since a large volume of data causes high storage, communication bandwidth, and energy usage [1]–[7]. There are two types of compression techniques, lossy and lossless. Lossless compression can generate the reconstructed datasets without losing data to its original form. On the other hand, lossy compression can generate the reconstructed datasets with the loss of some data to their original datasets. Various lossy compression algorithms have been proposed to meet applications' critical data management needs and can realize higher compression ratios than lossless ones. However, lossy compression faces the challenge of finding an optimal balance between data reduction and data fidelity, which is highly dependent on application domains and might affect the quality of analytic outcomes using the reconstructed data.

Transform-based lossy compressors for HPC datasets can also help reduce the size significantly. However, errors are hard to bound as they discard or quantize information in the transformed domain [1], [8]–[10]. Prior studies showed that transformation-based lossy compressors help minimize data reconstruction errors and are crucial for maintaining errors within a tolerable error bound [8], [11]. Furthermore, several researchers have studied the effect of lossy data compression on data fidelity and how it affects data analytics tasks, such as anomaly detection in streaming datasets. However, the relationship between the characteristics of HPC datasets and the behavior of transform-based lossy compressors is relatively unexplored. Thus, our motivation in this paper is to present a prediction model for dominant coefficients that transform-based compressors would capture, which are the critical factor for predicting compression ratios.

In this paper, we consider three transform-based lossy compressors, namely DCT-EC (Energy Compaction) [1], [8], DCT-K (Knee-point) [1], [12], and DCT-Z [13], and analyze the relationship between their compression performance in terms of dominant transform coefficients and data characteristics. All schemes are based on discrete cosine transform (DCT) and utilize block decomposition to capture spatial redundancy better and exploit parallelism. In detail, however, each scheme has a different compression process, especially in obtaining the information that needs to be kept in the highest precision, which ultimately dictates overall compression performance. We refer to that information as K-dominant coefficients. DCT-EC focuses on the K-dominant coefficients according to the fixed energy, and DCT-K obtains K-dominant coefficients using a knee-point detection mechanism. DCT-EC and DCT-K store data related to K-dominant coefficients during the compression process. Unlike DCT-EC and DCT-K, DCT-Z applies quantization to transformed datasets and obtains K-dominant coefficients outside specified error bounds to reduce data loss (i.e., errors).

Prior studies also indicate that the performance of lossy compressors depends on data as bit-plane coding needs to be applied differently [1], [6], [7], [10]. Using sampling mechanisms to predict compression ratios is one of the ways to exploit spatial or temporal data characteristics [14]–[17]. The transform-based compression could better reveal the relationship between achievable compression ratios and information loss, as low-frequency components carry more critical information than high-frequency ones. In other words, one can model the behavior of transform-based lossy compressors using how dominant coefficients compact and how they

978-1-6654-6337-9/22/$31.00 ©2022 IEEE
DOI 10.1109/DRBSD56682.2022.00013

relate to data characteristics at the block level. To verify the model for predicting compression ratios for DCT-EC, DCT-K, and DCT-Z, we use six production-level HPC datasets: eddy, vortex, sedov, cellular, rlds, and mrsos. We extract the statistical features of those datasets and label K-dominant coefficients for DCT-EC, DCT-K, and DCT-Z to conduct extensive evaluations of our prediction models.

We exploit the characteristics in block partitioned data in the original and DCT domain to capture the entire dataset's behavior and use them as input to prediction models. We evaluate five commonly-used regression machine learning models to predict the number of dominant coefficients that transforms-based lossy compressors preserve to balance compression ratios and error rates. Our experimental evaluation shows that our method can achieve 100% prediction accuracy in many cases of real scientific HPC data sets. Our results also demonstrate that the prediction model based on Random Forests (RF) presents its superior performance in terms of accuracy among the regression models we evaluated.

II. Related Work

One way to characterize lossy compressors is to accurately predict compression ratios using a small dataset instead of performing a trial-and-error approach, which is prohibitively expensive. Zemliachenko et al. [14] proposed a method to predict the compression of noisy images, the mean probability that absolute values of DCT coefficient amplitudes presumably could characterize the image, and also assumed that there is a relationship between the compression ratio and this parameter. Then the dependence of compression ratio on this parameter is obtained by curve fitting into a scatter-plot.

Lu et al. [15] proposed a sampling-based estimation method based on byte entropy, coreset size, and serial correlation coefficient to estimate the compression ratio of ZFP, ISABELA, and SZ. However, this method can only support Huffman encoding but not dictionary encoding. Moreover, it cannot estimate the compression ratio for SZ 2.0, which employs a new predictor.

Zhao et al. [17] also employ a sample engine that selects a small portion of the whole dataset by a uniform sampling method. The experiments show that under a small sampling rate of 8%, they can estimate compression ratio with only about 5% error in most cases.

Another prior study is from Zhang et al. [16], which employs a sampling strategy to estimate compression ratios for multi-staged lossy compressors. They first shrink the input dataset to a smaller dataset, randomly pick 3 (by default) subsets as sample data, then calculate the knee point for each subset and average them to get the knee point for the whole dataset. Doing this requires less calculation as it needs only three subsets instead of whole datasets. It also produces precise rate-distortion prediction, indicating that the prediction model works well with transformed domains.

III. Prediction of Compression Ratios

Transform-based lossy compressors employ discrete transforms, such as DCT, DWT, and Fast Walsh-Hadamard (FWHT), to decorrelate the original data [18]. Like the one used in [9], [19], due to its superior compaction capability and faster performance, we also adopt DCT-II, one of the most commonly used variants of DCT transforms. In transform-based lossy compression algorithms [20], unlike original signals, the transformed signals often contain *only* a few significant components [1], [9], [11], [21]. In other words, discrete transforms redistribute the energy contained in the signal and concentrate it into a small number of dominant coefficients (as low-frequency coefficients).

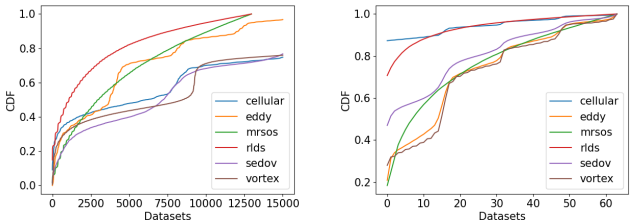

Fig. 1. Cumulative density function (CDF) of energy in DCT coefficients using (a) the entire data points and (b) the block of 64. The curves in (b) are based on averaged DCT coefficient values in all blocks.

Transform-based lossy compressor uses transformed coefficients to decide which information to keep or discard [1], [22]. In other words, the behaviors in transform coefficients rather than original data can effectively expose the importance of information. To demonstrate the effect of this relationship on data compression, Fig. 1 shows the CDF (cumulative density function) for the energy of DCT coefficients after applying DCT to the original HPC datasets, where each value is transformed into a DCT coefficient [22]. The energy in this paper is defined as a sum of the nonnegative value of the square root of the transformed DCT coefficients in sorted order. Since low-frequency DCT coefficients attain more energy (i.e., information), CDF curves climb quickly in the beginning and taper off as it includes more high-frequency coefficients, which are close to zero. As depicted in Fig. 1, each dataset shows different curvatures and saturation points, demonstrating each dataset's characteristics. Once the data is represented in the frequency domain, we can easily find the relationship between the percentage of informative DCT coefficients (i.e., low-frequency ones) and the amount of energy carried by them. Fig. 1a shows the CDF curves for each dataset using the entire data points, whereas Fig. 1b shows the CDF per block of 64. As we can see, unlike Fig. 1a, CDF curves in Fig. 1b (i.e., per block) are slightly faster to converge, but both show similar overall patterns. These CDF curvatures indicate that we can predict the data characteristic of the entire data points using the characteristics of data points in the partitioned blocks.

The above-mentioned unique features of discrete data transform inspired us to design a set of transform-based lossy compressors, DCT-EC, DCT-K, and DCT-Z. These lossy compressors have a different methodology for capturing dominant

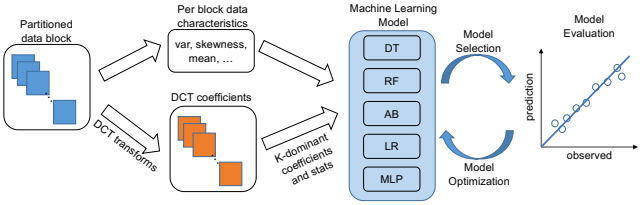

Fig. 2. An overview of the proposed approach to selecting the best-fit model for predicting K-coefficients, which directly affect compression ratios and error rates.

coefficients using the advantages of data transformation. Since the lossy compressors introduce the loss of data fidelity while reducing data, it faces a challenging understanding of the impact of lossy on HPC data management. The objective of this paper is to present a prediction model of compression ratios (in terms of the number of dominant coefficients to maintain specified accuracy) for the transform-based lossy compressor: DCT-EC, DCT-K, and DCT-Z. While there is a gap between the actual compression ratios and the number of dominant coefficients, the latter is much easier to extract. More importantly, it makes the relationship between error rates and compression ratios easier.

Fig. 2 shows the model for predicting compression ratios. To describe the model, let $X_{(t)}^{(i)}$ denote data points, where $1 \leq i \leq p$ and $1 \leq t \leq$ block size (BS), and i and t are block indexes and data point indexes in the block i, respectively. It also requires statistical features for each block, denoted as $\theta^{(i)}$, which characterizes data points $[X_{(1)}^{(i)}...X_{(BS)}^{(i)}]$. $\theta^{(i)}$ is the statistical characteristics of n data points in block i. Each block i includes datasets $X_{(t)}^{(i)}$ follows the condition in Equation 1.

$$X_{(t)}^{(i)} = \left\{ \begin{array}{l} X_{(t)}^{(1)}, \text{ if } 1 \leq t \leq BS, [\theta_{(1)}^{(1)}, ..., \theta_{(n)}^{(1)}] \\ X_{(t)}^{(2)}, \text{ if } 1 \leq t \leq BS, [\theta_{(1)}^{(2)}, ..., \theta_{(n)}^{(2)}] \\ \quad ... \\ X_{(t)}^{(p)}, \text{ if } 1 \leq t \leq BS, [\theta_{(1)}^{(p)}, ..., \theta_{(n)}^{(p)}] \end{array} \right\}, 1 \leq i \leq p. \tag{1}$$

We next define the compression ratio to predict, denoted as CR_i, and approximated data points for each block, denoted as AD_i. As shown in Equation 2 below, CR_i and AD_i are similar, except that AD_i includes additional information required for data reconstruction. We assume a strong relationship between $\theta^{(i)}$ (representing each block's data characteristics) and the compression ratio, CR_i, in each block as described in Equation 3, which we evaluate using several regression models.

$$AD_i(X_{(t)}^{(i)}) \approx CR_i, 1 \leq i \leq p. \tag{2}$$

$$CR_i \propto \theta^{(i)}. \tag{3}$$

Both transform-based lossy compressors and the model for predicting compression ratios aim to find K, which is the number of significant coefficients in the transformed domain required to store outcome X_k out of the original dataset X to reconstruct data within a certain tolerance. We then characterize the K needed to approximate X and the reconstructed $R(X_k)$. The rate-distortion of X_k measures $ER_k|X, R(X_k)|$. $ER_k|X, R(X_k)|$ means the error rate between original datasets X and reconstructed datasets $(R(X_k))$ from lossy compressed datasets (X_k).

We choose the following statistical features (denoted as $\theta^{(i)}$) to characterize datasets and related to the transformation compression:

- Descriptive features: we calculate variance, mean, median, range, diff, skewness, and kurtosis of datasets. Skewness is a measure of data asymmetry around the mean value. Normal distribution, which is symmetric around its mean, gives zero skewness. Negative skewness values mean more data are scattered to the left of the mean, whereas positive skewness values mean more data are scattered to the right. The kurtosis measurement indicates how much the distribution tends to be outliers. As the kurtosis of any normal distribution is 3, distributions with a kurtosis higher than 3 are more outlier-prone. On the other hand, the distributions of a kurtosis lower than 3 are less outlier-prone. We also measure diff defined as follows: diff$(t) = X_{t+1}^i - X_t^i, 1 \leq t \leq BS$ for block i. We calculate the standard deviation of diff(x).
- Stationarity features: stationarity means the properties, like mean, variance, and autocorrelation structure, do not change over time. Therefore, determining whether a partitioned block is stationary is crucial to making accurate predictions. We use two statistical tests to check the stationarity of a dataset – the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. A key difference from the ADF test is that the null hypothesis of the KPSS test is that the block is stationary. So the interpretation of the probability value (p-value) is just the opposite of each other. In other words, if the p-value is above the 0.05 level, then the tested data is stationary. Whereas in the ADF test, it means non-stationary.
- Transform coding features: we also calculate the std(dct(X_i) and max(dct(X_i)) in the given block X_i.

Selecting an appropriate algorithm is essential when the training data is ready to use after the preprocessing, which involves balancing the class distribution of given data, normalization, and missing value analysis. However, all learning algorithms do not perform equally well when averaged over all possible data sets. In other words, looking for a general and superior algorithm to other algorithms is not feasible. Therefore, we conduct an extensive evaluation using several machine learning algorithms to predict compression ratios (later in Section IV-D).

The following subsections discuss how each lossy compressor we evaluate extracts K in detail.

A. DCT-EC

DCT-EC transforms partitioned data blocks into sparse representations using DCT. To illustrate DCT-EC in detail,

let us consider x and $\hat{x}$ to represent the original data and corresponding transformed components, respectively [22]. To model the correlation between the transformed coefficients and energy (or information) represented among them, let us further define $\hat{x}_i$, which denotes transformed components in a block size of N at given block i: $\hat{x}_i = \{\hat{x}_{i,1}, \hat{x}_{i,2}, ...\hat{x}_{i,N}\}$. $EC(\hat{x}_{i,k})$ is formulated as the energy concentration (EC) contained in the number of DCT coefficients, denoted as K, of the entire transformed components ($\hat{x}_i$), which is calculated as:

$$EC(\hat{x}_{i,k}) = \frac{\sum_{n=1}^{k} e(\hat{x}_{i,n})^2}{\sum_{n=1}^{N} e(\hat{x}_{i,n})^2}, n = 1, 2, ..., N, i \leq N. \quad (4)$$

Using Equation 4 above, DCT-EC finds the dominant K required to represent δ amount of the energy [19] in terms of energy compaction. K refers to the number of dominant coefficients to represent block $\hat{x}_i$ or x_i.

B. DCT-K

Another compression scheme we consider automatically determines compression factors (i.e., dominant coefficients) based on the curvature of energy compaction attained by the transformed coefficients. We adopt the Kneedle algorithm [12] on the CDF (cumulative density function) of the coefficients' energy compaction [1] to determine K. Specifically, we first fit CDF into a smoothing spline to preserve the overall behavior of the energy distribution. Then we normalize the points in the best-fit curve to the unit square as a preprocessing step to eliminate anomalies (which can make the analysis more complicated). Next, we find the knee points (in K) from the normalized curve, generally the point of maximum curvature of the normalized curve. In other words, the point is the local maxima, which depicts the maximum distance between the normalized curve and the line $y = x$; mathematically, it is a function of its first and second derivatives. We use a 1D interpolation function for obtaining smoothing curves, while a more sophisticated one, such as a polynomial interpolation function, can also be applied.

C. DCT-Z

DCT-Z [9] is another transform-based lossy compression method but with an error boundness. In DCT-Z, the first DCT coefficient ($\hat{x}_{i,1}$) in $\hat{x}_i$ is the most informative coefficient (i.e., DC coefficient, which contains zero frequency), and the remaining coefficients as the AC coefficients (contain non-zero frequencies). Similar to JPEG, DCT-Z utilizes the 8x8 macroblock, which could capture 8×8 in 2D or 4×4×4 in 3D, a common data partition used in many scientific applications. Once the partitioned block is transformed using DCT, it aggregates the DC coefficient, i.e., $\hat{x}_{i,1}$, from each block and organizes them based on the block sequence order. Therefore, the aggregated DC coefficients represent low-frequency coefficients for the entire data. The remaining AC coefficients, $\{\hat{x}_{i,2}, ...\hat{x}_{i,N}\}$, are considered as the high-frequency coefficients. While different block sizes affect data precision, we found the default 64 generated the best compression quality in most cases. The block size of 64 also enables easier parallelization of DCT transforms. While one could employ recursive DCT, additional indexing and ordering coefficients incurred during compression result in lower compression ratios. More details about DCT-Z can be found in [13].

DCT-Z implements the error boundness through an adjustable quantizer. The quantizer begins with the user-defined (relative) error bound, denoted as P, and a total number of bins, C. Using P and C, it defines a global bound (GP) as $[-P*C, P*C]$. For example, if P is 1e-3 and C is 256 (i.e., 1 byte index for bin indexes), then GP is $[-0.256, 0.256]$. Each bin's center value will be approximated values for all AC coefficients belonging to the same bin. Once GP is calculated, DCT-Z determines how many AC coefficients are within GP or not. If AC coefficients are within GP, then the maximum errors due to approximation are within the user-specified error bound, P. For AC coefficients outside GP, DCT-Z saves them as the exact values to ensure the user-defined error bound. One could also apply an extra truncation to improve the compression ratio on AC components out of GP. Since AC components that need to be saved as exact values dictate the compression factor, we calculate K-dominants the same as coefficients outside the error bound P.

IV. Evaluations

A. Setup

1) Datasets: We use six datasets from three production-level HPC applications. Specifically, we use sedov and cellular from FLASH [23], [24], Vortex and Eddy from Nek5000 [25], [26], and mrsos and rlds from CMIP5 [27]. Prior studies showed that these datasets worked well with transform-based compressors, but we expect our framework will work with other datasets, like the ones in SDRBench [28], which we plan to explore in our future work. Table I shows the statistical properties of the original datasets in variance, skewness, and kurtosis. We measure each statistics per block, and Table I shows the mean value for all blocks. Skewness is a measure of data asymmetry around the mean value. As shown in Table I, sedov has the highest kurtosis value among all datasets, which means it is more outlier prone. In the case of variance, diff-std, and range(max-min), mrsos has the highest values. In the case of transform coding, rlds has a higher value than others – consequently, the data characteristics in Table I would impact K presented in Table II.

As described in the previous section, the method of determining the K for prediction differs among DCT-EC, DCT-K, and DCT-Z. K of DCT-EC is the same as the K-dominant coefficients and the knee-point in DCT-K. In DCT-Z, the K-dominant coefficients are the same as those outside the error bound. We note that the predicted K is the same as compression ratios for DCT-EC and DCT-K as both truncate non-significant components (i.e., $N - K$). On the other hand, the predicted K is not the actual compression ratio for DCT-Z because it maintains non-significant components through quantization.

TABLE I
THE CHARACTERISTICS (IN MEAN) OF EVALUATED DATASET PER BLOCK.

Data	Var	Mean	Median	Skewness	Kurtosis	ADF	KPSS	DCT-max	DCT-std	diff-std	range
cellular	0.04	2.21	2.21	-0.11	1.82	0.63	0.1	17.69	2.2	0.09	0.26
eddy	0.33	0.0	0.02	-0.15	-0.68	0.55	0.05	3.54	0.8	0.37	1.8
mrsos	0.84	0.77	0.55	1.19	2.74	0.26	0.09	6.24	1.09	0.41	2.4
rlds	0.13	2.86	2.86	-0.02	-0.21	0.26	0.08	22.91	2.85	0.12	1.18
sedov	0.21	1.0	0.93	0.91	3.62	0.6	0.06	8.0	1.04	0.21	0.94
vortex	0.34	0.17	0.1	0.01	0.24	0.49	0.09	3.52	0.72	0.36	1.08

TABLE II
THE AVERAGE VALUES OF K FOR EACH COMPRESSOR.

Data	EC (95)	EC (99)	EC (999)	K	Z(1e-3)	Z(1e-4)	Z(1e-5)
cellular	1.11	1.85	3.24	9.8	3.06	14.21	39.82
eddy	10.4	22.18	40.49	9.12	17.8	53.11	62.83
mrsos	15.03	28.26	42.01	10.75	28.09	45.47	48.11
rlds	1.17	2.34	9.19	8.98	9.68	43.74	59.8
sedov	4.74	11.62	21.52	7.75	10.07	25.69	32.2
vortex	6.7	14.26	25.06	9.81	10.78	32.96	53.65

B. Performance Metrics

Let $x = \{x_1, x_2, x_3, ..x_N\}$ be the raw HPC datasets, N be the number of data points in raw datasets (x), and n be the number of blocks calculated by $\frac{N}{\text{block size}}$, where block size is 64 in this paper.

- Compression Ratio for DCT-EC, DCT-K, and DCT-Z where $|D|$ is the number of D, $|k'|$ is the number of K-dominant coefficients, is given by: $CR_K = \frac{|x|-|k'|}{|x|}$
- Let $x = x_1, x_2, x_3, ..x_N$ be the original data and $\hat{x} = \hat{x}_1, \hat{x}_2, \hat{x}_3, ...\hat{x}_N$ be the reconstructed data. Then, we measure the peak signal-to-noise ratio (PSNR), a commonly used average error metric, especially in visualization [29], which is calculated as follows:

$$PSNR = 20log_{10}(\frac{Max(x) - Min(x)}{RMSE(x, \hat{x})}).$$

- Accuracy of the prediction is defined as R^2:

$$Accuracy(y_t, y'_t) = 1 - \frac{\sum_{t=1}^{n} e_t^2}{\sum_{t=1}^{n}(y_t - \overline{y_t})^2},$$

where $e_t = (y_t - y'_t)$, y_t is actual compression ratios y_t and y'_t is predicted values, both are represented in K values. The best possible score is 1.0, i.e., 100% accurate prediction. The score can also be less than zero because the model can be arbitrarily worse.

- Mean square error of the prediction is defined as:

$$MSE(y_t, y'_t) = \frac{1}{N}\sum_{t=1}^{n}(e_t)^2.$$

C. Compression Performance

Tables III and IV show compression ratios and PSNR using DCT-EC (with the fixed information compaction rate of 95%, 99%, and 99.9%), DCT-K, and DCT-Z. In DCT-Z, we use three specified error bounds (P): 1e-3, 1e-4, and 1e-5. All schemes apply transforms per every 64 data points, i.e., the block size is 64. Given the number of data points per dataset, the number of blocks after partitioning ranges from 203 to 579. Typically, blocks with higher variance require more components to approximate data maintained by the same amount of energy, resulting in a lower approximation ratio. For instance, mrsos has slightly higher K values than other data sets in Table II. Therefore, the std and diff_std values in mrsos are higher than other data sets in Table I. In the case of DCT-Z, when the error bound is the highest value (i.e., 1e-3), it shows the highest compression ratios. Table IV shows that higher compression leads to higher error rates with lower PSNRs.

TABLE III
THE EVALUATED DATASETS AND THEIR COMPRESSION RATIOS (%).

Data	DCT-EC			DCT-K	DCT-Z		
	95	99	999		1e-3	1e-4	1e-5
celllular	97.27	97.11	94.94	84.68	98.09	96.11	94.13
eddy	83.76	65.34	36.74	85.75	84.71	69.33	63.24
mrsos	76.51	55.84	34.36	83.21	77.17	70.50	70.50
rlds	98.17	96.35	85.64	85.97	83.02	68.15	63.90
sedov	92.59	81.84	66.38	87.89	91.30	86.05	83.33
vortex	89.54	77.71	60.85	84.67	92.23	83.55	77.63

TABLE IV
THE EVALUATED DATASETS AND THEIR ERROR RATE (PSNR).

Data	DCT-EC			DCT-K	DCT-Z		
	95	99	999		1e-3	1e-4	1e-5
cellular	25.4	30.65	36.53	32.71	75.62	95.61	116.88
eddy	28.33	35.05	44.91	28.43	79.75	103.59	127.87
mrsos	23.36	30.08	40.23	22.54	82.66	111.25	127.89
rlds	21.16	24.54	32.17	31.38	76.78	100.93	126.15
sedov	25.91	32.35	42.27	28.35	81.75	104.14	125.31
vortex	28.48	35.07	100	30.03	81.26	102.99	125.00

D. Prediction Performance

We use the following regression methods in our evaluation: Decision Tree (DT) [30], Adaptive Boosting (AB) [31], Random Forest (RF) [32], Linear Regression (LR), and Multi-layer Perceptron (MLP) [33].

- DT is a popular nonparametric supervised learning method to solve various real-world problems. It operates by repeatedly dividing a feature space of a given dataset using the Gini index and Entropy.
- AB is also an improvement of DT to fit a sequence of weak learners (i.e., models that are only slightly better than a random guess, such as small decision trees) on repeatedly modified versions of the data. Then, the predictions from all of them are combined through a

weighted majority vote (or sum) to produce the final prediction.

- RF is an ensemble model correlated DTs to solve the vulnerabilities mentioned above by combining multiple DTs.
- LR fits a linear model with coefficients to minimize the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation.
- MLP is a mathematical model inspired by the study of the brain. It has a connected set of artificial neurons consisting of the input layer, one or more hidden layers, and an output layer.

As previously described in Table IV, the K values increase with tighter constraints in DCT-EC and DCT-Z. For example, DCT-EC with 0.999 requires a higher K than DCT-EC with 0.99. Similarly, DCT-Z with 1e-5 requires a higher K than DCTZ-Z with 1e-3 as 1e-5 is a tighter error bound than 1e-3. Note that DCT-K automatically determines K based on the detected knee points in the CDF curve. We predict K to represent the compression ratios using the characteristics of the data. We utilize machine learning libraries from scikit-learn [34] to train and evaluate the five machine learning algorithms we evaluate. We set 80% of the data for training and the remaining 20% for the test. Tables V, VI and VII show the model's prediction performance on compression ratios for DCT-Z, DCT-EC, and DCT-K, respectively. We evaluate performance according to fixed energy (EC: 0.95, 0.99, and 0.999, Z: 1e-3, 1e-4, and 1e-5). As we can see, RF performs better than other algorithms in prediction accuracy for all datasets. In conjunction with the compression ratios, the observed data characteristics show a clear correlation between those two. In the case of DCT-Z, the accuracy of eddy in the MLP model is less than zero because the model could not determine the compression ratio based on the data loss of the application effectively. Based on these results for model selection, we can conclude that RF works best to work with reconstructed data from lossy compression.

We next perform optimization for the selected model, RF. The parameters of the RF model are optimized through a cross-validated grid search. Specific optimized parameters we obtained are as follows:

- The optimized number of features is 2 from (2, 4, 6, 8, 10, 12).
- The optimized number of estimators is 3 from (3, 10, 30, 100).

TABLE V
EXPERIMENTAL RESULTS FOR MODEL SELECTION. DCT-Z

Data	GP	DT		AB		RF		LR		MLP	
		Acc	MSE	Acc	MSE	Acc	MSE	Acc	MSE	Acc	MSE
cellular	1e-3	**1.0**	**0.01**	**1.0**	**0.0**	**1.0**	**0.0**	0.95	2.23	**1.0**	0.09
	1e-4	0.98	3.94	**1.0**	0.66	**1.0**	**0.01**	0.66	76.07	0.94	13.41
	1e-5	0.96	6.37	**1.0**	0.51	**1.0**	**0.08**	0.39	95.89	0.64	56.48
eddy	1e-3	0.94	5.32	0.99	0.56	**1.0**	**0.18**	0.8	19.45	0.94	6.0
	1e-4	0.95	5.77	0.99	1.14	**1.0**	**0.34**	0.56	48.75	0.67	36.86
	1e-5	0.84	0.43	0.94	0.16	**0.98**	0.06	0.36	**1.77**	-4.4	15.02
mrsos	1e-3	0.98	4.8	0.99	1.38	**1.0**	**0.99**	0.96	9.12	0.96	9.44
	1e-4	**1.0**	1.45	**1.0**	0.71	**1.0**	**0.23**	0.91	48.46	0.91	44.91
	1e-5	**1.0**	0.18	**1.0**	**0.17**	0.99	7.83	0.81	98.47	0.8	103.9
rlds	1e-3	0.88	1.96	0.97	0.49	**0.99**	**0.22**	0.8	3.3	0.86	2.35
	1e-4	0.92	12.48	0.99	2.27	**0.99**	**1.04**	0.75	40.23	0.74	42.23
	1e-5	0.97	2.04	0.99	1.07	**0.99**	**0.59**	0.38	49.09	0.0	78.45
sedov	1e-3	0.98	6.21	0.99	2.64	**0.99**	**1.68**	0.8	50.63	0.94	14.57
	1e-4	0.98	7.93	0.99	2.42	**1.0**	**1.23**	0.66	151.03	0.84	71.42
	1e-5	0.99	1.74	**1.0**	0.92	**1.0**	**0.25**	0.29	191.46	0.47	144.03
vortex	1e-3	**1.0**	0.12	**1.0**	0.05	**1.0**	**0.0**	0.97	2.93	**1.0**	**0.16**
	1e-4	0.99	3.5	**1.0**	1.0	**1.0**	**0.01**	0.81	45.78	0.99	3.47
	1e-5	0.95	5.16	0.99	1.44	**1.0**	**0.01**	0.65	35.43	0.85	14.95

TABLE VI
EXPERIMENTAL RESULTS FOR MODEL SELECTION.

Data	EC	DT		AB		RF		LR		MLP	
		Acc	MSE	Acc	MSE	Acc	MSE	Acc	MSE	Acc	MSE
cellular	0.95	**1.0**	**0.0**	**1.0**	**0.0**	**1.0**	**0.0**	0.91	0.02	0.99	**0.0**
	0.99	**1.0**	**0.0**	**1.0**	**0.0**	**1.0**	**0.0**	0.8	1.9	**1.0**	0.02
	0.999	**1.0**	0.01	**1.0**	**0.0**	**1.0**	**0.0**	0.88	6.53	**1.0**	0.07
eddy	0.95	0.96	2.1	**0.99**	0.39	**0.99**	**0.26**	0.73	13.31	0.94	3.16
	0.99	0.88	14.67	0.99	1.59	**1.0**	**0.28**	0.71	35.77	0.91	10.62
	0.999	0.93	10.24	0.99	1.61	**1.0**	**0.36**	0.65	50.14	0.84	22.63
mrsos	0.95	0.92	7.64	**0.99**	**0.96**	**0.99**	1.04	0.88	11.32	0.92	7.02
	0.99	0.87	17.78	0.98	2.31	**0.99**	**1.42**	0.8	28.03	0.89	15.4
	0.999	0.92	14.1	0.98	2.91	**0.99**	**1.92**	0.71	50.07	0.84	26.72
rlds	0.95	**1.0**	**0.0**	**1.0**	**0.0**	0.97	0.01	0.8	0.06	0.84	0.05
	0.99	0.85	0.44	**0.99**	**0.03**	0.98	0.05	0.77	0.66	0.96	0.12
	0.999	0.93	2.2	**0.99**	**0.37**	**0.99**	0.39	0.92	2.5	0.95	1.34
sedov	0.95	0.94	4.27	0.98	1.38	**0.99**	**0.39**	0.55	31.76	0.91	6.64
	0.99	0.89	18.62	0.97	5.49	**0.99**	**0.95**	0.51	79.58	0.77	37.0
	0.999	0.92	18.04	0.97	7.67	**0.99**	**1.48**	0.34	147.79	0.56	99.05
vortex	0.95	0.98	0.71	**1.0**	0.07	**1.0**	**0.03**	0.88	3.6	0.99	0.2
	0.99	0.99	1.46	**1.0**	0.24	**1.0**	**0.0**	0.92	10.22	**1.0**	0.59
	0.999	0.98	2.96	**1.0**	0.49	**1.0**	**0.01**	0.85	29.73	0.99	1.75

TABLE VII
EXPERIMENTAL RESULTS FOR MODEL SELECTION. DCT-K.

Data	DT		AB		RF		LR		MLP	
	Acc.	MSE	Acc.	MSE	Acc.	MSE	Acc.	MSE	Acc.	MSE
cellular	0.8	3.34	0.95	0.76	**1.0**	**0.02**	0.56	7.2	0.84	2.66
eddy	0.73	1.8	0.97	0.21	**0.99**	**0.05**	0.6	2.69	0.82	1.23
mrsos	0.75	4.85	0.95	0.87	**0.97**	**0.55**	0.77	4.51	0.82	3.52
rlds	0.75	1.47	0.92	0.45	**0.97**	**0.17**	0.59	2.35	0.67	1.92
sedov	0.91	4.68	0.97	1.55	**0.99**	**0.68**	0.82	9.85	0.9	5.4
vortex	0.99	0.17	0.99	0.11	**1.0**	**0.04**	0.89	1.31	0.98	0.27

V. CONCLUSIONS

This paper analyzes three transform-based lossy compression techniques for HPC datasets: DCT-EC, DCT-K, and DCT-Z. Our motivational analysis showed that the behavior exhibited in the partitioned block level in the transformed domain could represent that of the entire dataset. We exploit this behavior to build a model that maps block-level data characteristics from various perspectives to predict the number of dominant coefficients for different transformation-based lossy compressors. Our evaluation results using several real-world HPC datasets showed that the higher the data variance, the lower the number of dominant coefficients but with some differences among the compressors. Especially, DCT-EC and DCT-Z require more DCT components to maintain the same amount of energy or error-bound when datasets exhibit higher variance. In other words, higher randomness would result in a lower compression ratio. We also evaluate the accuracy and error rate of five machine learning models. Our evaluation results indicated that RF is among the five regression models that achieve the highest prediction accuracy for all datasets we evaluated. We expect that the proposed model to effectively predict the number of dominant coefficients could help quantify how transform-based lossy compressors impact overall compression ratios and a range tolerable by the application with reliable precision.

ACKNOWLEDGMENT

This work is supported by the Korea Innovation Foundation (INNOPOLIS) grant funded by the Korea government (MSIT) (2020-DD-UP-0278). This material is also in part based upon work supported by the National Science Foundation under Grant No. 1751143. The Titan X Pascal used for this research was donated by the NVIDIA Corporation.

REFERENCES

[1] J. Zhang, A. Moon, X. Zhuo, and S. W. Son, "Towards Improving Rate-Distortion Performance of Transform-Based Lossy Compression for HPC Datasets ," in *IEEE HPEC*, 2019.

[2] Z. Chen, S. W. Son, W. Hendrix, A. Agrawal, W.-k. Liao, and A. Choudhary, "NUMARCK: Machine Learning Algorithm for Resiliency and Checkpointing," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 733–744.

[3] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W.-k. Liao, and A. Choudhary, "Data Compression for the Exascale Computing Era - Survey," *Supercomputing Frontiers and Innovations*, vol. 1, no. 2, p. 76–88, Sep. 2014. [Online]. Available: https://superfri.susu.ru/index.php/superfri/article/view/13

[4] J. Zhang, A. Moon, X. Zhuo, and S. W. Son, "Towards improving rate-distortion performance of transform-based lossy compression for hpc datasets," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1–7.

[5] J. Zhang, X. Zhuo, A. Moon, H. Liu, and S. W. Son, "Efficient encoding and reconstruction of hpc datasets for checkpoint/restart," in *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*, 2019, pp. 79–91.

[6] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, "TTHRESH: Tensor Compression for Multidimensional Visual Data," *IEEE Transaction on Visualization and Computer Graphics*, vol. 26, pp. 2891–2903, 2019.

[7] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky, "Multilevel Techniques for Compression and Reduction of Scientific Data–the Univariate Case," *Comput. Vis. Sci.*, vol. 19, no. 5–6, p. 65–76, dec 2018. [Online]. Available: https://doi.org/10.1007/s00791-018-00303-9

[8] A. Moon, J. Kim, J. Zhang, and S. W. Son, "Lossy Compression on IoT Big Data by Exploiting Spatiotemporal Correlation," in *2017 IEEE HPEC*, 2017, pp. 1–7.

[9] J. Zhang, X. Zhuo, A. Moon, H. Liu, and S. W. Son, "Efficient Encoding and Reconstruction of HPC Datasets for Checkpoint/Restart," in *Symposium on Mass Storage Systems and Technologies (MSST)*, 2019.

[10] P. Lindstrom, "Fixed-Rate Compressed Floating-Point Arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.

[11] A. Moon, J. Kim, J. Zhang, H. Liu, and S. W. Son, "Understanding the Impact of Lossy Compression on IoT Smart Farm Analytics," in *2017 IEEE Big Data*, 2017, pp. 4602–4611.

[12] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW '11. USA: IEEE Computer Society, 2011, p. 166–171. [Online]. Available: https://doi.org/10.1109/ICDCSW.2011.20

[13] https://https://github.com/swson/DCTZ.

[14] A. N. Zemliachenko, S. Abramov, V. V. Lukin, B. Vozel, and K. Chehdi, "Compression ratio prediction in lossy compression of noisy images," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 3497–3500.

[15] T. Lu, Q. Liu, X. He, H. Luo, E. Suchyta, J. Choi, N. Podhorszki, S. Klasky, M. Wolf, T. Liu, and Z. Qiao, "Understanding and modeling lossy compression schemes on hpc scientific data," in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2018, pp. 348–357.

[16] J. Zhang, J. Chen, X. Zhuo, A. Moon, and S. W. Son, "Dpz: Improving lossy compression ratio with information retrieval on scientific data," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, 2021, pp. 320–331.

[17] K. Zhao, S. Di, X. Liang, S. Li, D. Tao, Z. Chen, and F. Cappello, "Significantly improving lossy compression for hpc datasets with second-order prediction and parameter optimization," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 89–100. [Online]. Available: https://doi.org/10.1145/3369583.3392688

[18] A. Moon, K. Y. Moon, and S. W. Son, "Microcilmate-Based Predictive Weather Station Platform: A Case Study for Frost Forecast," in *2017 IEEE High Performance Extreme Computing Conference*, 2017.

[19] A. Moon, J. Kim, J. Zhang, and S. W. Son, "Evaluating Fidelity of Lossy Compression on Spatiotemporal Data from an IoT Enabled Smart Farm," *Computers and Electronics in Agriculture*, vol. 154, pp. 304–313, Nov. 2018.

[20] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications," in *IEEE Access*, 2018, pp. 4875–4894.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[22] M. A. Razzaque, C. J. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, p. 5, 2013.

[23] Flash Center for Computational Science, "FLASH User's Guide: Version 4.4," http://flash.uchicago.edu/site/flashcode/user_support/flash4_ug_4p4.pdf.

[24] B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo, "FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes," *The Astrophysical Journal Supplement Series*, vol. 131, no. 1, p. 273, 2000. [Online]. Available: http://stacks.iop.org/0067-0049/131/i=1/a=273

[25] P. Fischer, J. Lottes, S. Kerkemeier, O. Marin, K. Heisey, A. Obabko, E. Merzari, and Y. Peet, "Nek5000 User Documentation," http://nek5000.github.io/NekDoc/Nek_users.pdf, Argonne National Laboratory, Tech. Rep. ANL/MCS-TM-351, 2015.

[26] "Nek5000 – A Spectral Element code for CFD," https://nek5000.mcs.anl.gov/.

[27] K. E. Taylor, R. J. Stouffer, and G. A. Meehl, "An Overview of CMIP5 and the Experiment Design," *Bulletin of the American Meteorological Society*, vol. 93, no. 4, pp. 485–498, 2012.

[28] K. Zhao, S. Di, X. Lian, S. Li, D. Tao, J. Bessac, Z. Chen, and F. Cappello, "Sdrbench: Scientific data reduction benchmark for lossy compressors," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 2716–2724.

[29] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly Improving Lossy Compression for Scientific DataSets Based on Multidimensional Prediction and Error-Controlled Quantization," in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2017, pp. 1129–1139.

[30] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers-a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 4, pp. 476–487, 2005.

[31] R. E. Schapire and Y. Freund, "Boosting: Foundations and algorithms," *Kybernetes*, 2013.

[32] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.