Anomaly Detection in Scientific Datasets using Sparse Representation

Aekyeung Moon
Minjun Kim
akmoon@etri.re.kr
minjunkim@etri.re.kr
Electronics and Telecommunication Research Institute
Daegu, South Korea

Jiaxi Chen
Seung Woo Son
jiaxi_chen@student.uml.edu
seungwoo_son@uml.edu
University of Massachusetts Lowell
Lowell, MA, USA

ABSTRACT

As the size and complexity of high-performance computing (HPC) systems keep growing, scientists' ability to trust the data produced is paramount due to potential data corruption for various reasons, which may stay undetected. While employing machine learning-based anomaly detection techniques could relieve scientists of such concern, it is practically infeasible due to the need for labels for volumes of scientific datasets and the unwanted extra overhead associated. In this paper, we exploit spatial sparsity profiles exhibited in scientific datasets and propose an approach to detect anomalies effectively. Our method first extracts block-level sparse representations of original datasets in the transformed domain. Then it learns from the extracted sparse representations and builds the boundary threshold between normal and abnormal without relying on labeled data. Experiments using real-world scientific datasets show that the proposed approach requires 13% on average (less than 10% in most cases and as low as 0.3%) of the entire dataset to achieve competitive detection accuracy (70.74%–100.0%) as compared to two state-of-the-art unsupervised techniques.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**.

KEYWORDS

Anomaly Detection, Sparsity Profile, Discrete Cosine Transform, Unsupervised Learning, Scientific Datasets

ACM Reference Format:

Aekyeung Moon, Minjun Kim, Jiaxi Chen, and Seung Woo Son. 2023. Anomaly Detection in Scientific Datasets using Sparse Representation. In *Proceedings of the First Workshop on AI for Systems (AI4Sys '23), June 20, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3588982.3603610

1 INTRODUCTION

The massive volumes of datasets generated by modern scientific models and simulations of significant importance can help scientific discoveries reach better and faster decisions if scientists leverage the datasets correctly. However, the potential compromise in the produced datasets, in particular, due to silent data corruption may stay undetected and ultimately can adversely affect the integrity of scientific interpretation [11, 12, 14, 17, 21, 22, 28–30, 33–35, 38, 39, 41]. For instance, Jaguar, a petascale supercomputer at ORNL, suffers a double-bit memory error once every 24 hours [17]. An efficient and effective anomaly detection empowers scientists to take timely actions to correct anomaly situations [4, 5, 23]. For example, the detector presented in [14] dynamically predicts the value for each data point at each time step and compares the observed value with a normal value range. They check every data point by injecting the errors with different bitflips in binary notations.

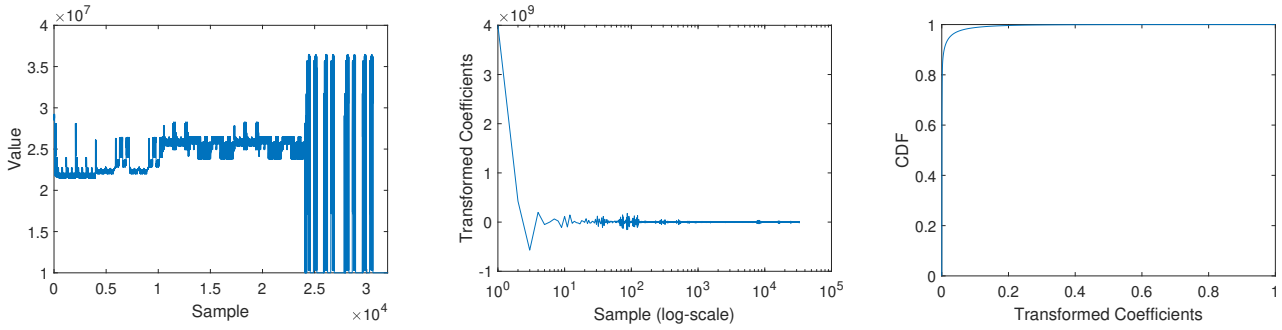
Anomaly detection has traditionally relied on the human expert's knowledge, like domain scientists who define the anomaly rule and deploy the defined rule-based anomaly detection algorithms to the target scientific datasets [2, 20, 37]. However, obtaining large-scale datasets with proper labels for supervised ML-based anomaly detection models and the unpredictability combined with the growing data size of HPC systems make effective use of anomaly detection challenging [14, 31, 37].

Our anomaly detection model combines the advantage of the sparse representation and anomaly detection model in an unsupervised manner. We exploit data transformation to obtain sparse representations at the block level as approximation through transforms can easily expose intrinsic features between *normal* and *abnormal* [26, 27]. We evaluate the proposed approach with six scientific datasets produced from three production-level simulation codes, FLASH [16], CMIP5 [24], and Nek5000 [15]. We compare the effectiveness of the proposed approach against two state-of-the-art unsupervised anomaly detection techniques, local outlier factor (LoF) and autoencoder (AE), in terms of detection performance with various degrees of injected errors (1%, 5%, and 10%). The injected error is sufficiently trivial, as measured mean absolute error (MAE) ranges from 1E-3 to 1E-1 to reflect realistic scenarios where those injected errors may stay undetected.

The experimental results show that the proposed approach can: 1) provide a systematic anomaly injection module for testing various contamination rates; 2) detect the actual abnormalities correctly, 76.41%–93.04% of point anomalies and 73.79%–91.30% of collective anomalies, respectively, when the corruption rate is 1% for the evaluated datasets; 3) achieve competitive approximation, 53.92%-99.81%, with less influence on errors from approximation for the entire evaluated datasets.

AI4Sys '23, June 20, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0161-0/23/06…$15.00
https://doi.org/10.1145/3588982.3603610



(a) Original (b) Transformed coefficients (c) CDF

Figure 1: The transformed representations of scientific datasets. (a) Original Cellular datasets. (b) Transformed coefficients (where the x-axis is in log scale). (c) CDF in the sequences of transformed (normalized) DCT coefficients.

2 PRELIMINARIES

2.1 Sparsity

As the amount of data produced by modern scientific simulation codes reaches the exabyte range, data reduction techniques are inevitable to lessen the volume of data and overhead by exploiting potential *redundancy* in scientific datasets [40]. We utilize this sparsity in scientific datasets to characterize potential data anomalies. For example, Figure 1a shows the original dataset from the Cellular Nuclear Burning problem in FLASH. Figure 1b shows the distribution of coefficients (in the spatial domain) after applying a discrete cosine transform (DCT) to the original datasets (Figure 1a). The x-axis label is in a log scale to emphasize that a small number of transformed components account for most of the information (signal amplitudes). Figure 1c illustrates such a high concentration of information through the cumulative distribution function (CDF) for DCT coefficients; DCT coefficients containing 99%-percentile are only 0.3% (110 out of 32,768) of the entire Cellular dataset. While the sparsity level varies among datasets, one can efficiently explore properties of scientific datasets after data transforms with a suitable basis [40, 42]. Among various transforms, such as DCT, Fourier, or Wavelet basis, DCT or similar variants, in particular, generates sparse representation effectively compared to other equivalent signal transformation methods [40].

2.2 Anomaly Detection

Anomaly detection is finding the patterns of various anomaly categories (such as point and collective) in datasets whose behavior is not as expected [10]. A typical output of anomaly detection is a trained binary classifier, which reads a new data entry as the input and outputs a hypothesis (1 or 0 with some confidence) for the data point's abnormality. These techniques, which have been extensively studied, range from training the detection algorithm using completely unlabelled data to having an organized dataset with entries labeled normal or abnormal and to those that rely only partially on external input [19]. For example, in [36], they collect several measurements through a monitoring infrastructure to classify the behavior of HPC systems using the collected statistical features. This classifier is based on supervised ML algorithms and decides the class (normal and abnormal).

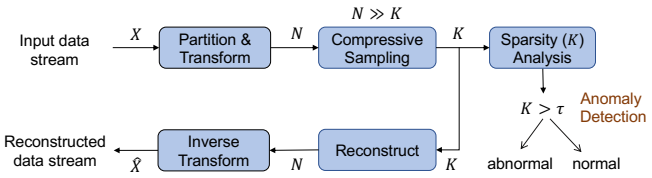


Figure 2: Overview of system architecture for learning sparse representations and detecting anomalies.

While prior studies showed that supervised learning-based approaches could generate promising results, this is only viable in cases where labeled data are abundant. However, there are few or no labeled examples of anomalous behavior in real-world scenarios. Furthermore, in many cases, it is infeasible to label every scientific dataset manually [6]. On the other hand, unsupervised algorithms detect anomalies solely based on the intrinsic properties of the unlabeled data points [3, 7, 9]. For example, the autoencoder in [7] and console logs generated by computing nodes in [13] are adopted to detect anomalies. However, these prior studies mainly focus on the anomaly of system behavior in the computing nodes (i.e., network contention or memory leak), not the data anomaly we focused on in this paper. Unlike ML-based approaches, an alternate solution, ABFT [12], could also detect anomalies accurately with less overhead. However, ABFT is always designed for a particular algorithm, requiring application modification or customization.

3 PROPOSED APPROACH

Figure 2 shows an overview of our methodology, where we learn the boundary threshold (τ) while using compressive sampling to classify whether blocks (block size is 100 in this paper) contain anomalies. In the detection phase, we investigate raw anomaly scores using only the K-dominant coefficients derived from SR and τ to classify whether a block contains anomalous data points. The evaluation process includes an error-injecting module to validate the effectiveness of the proposed model by comparing it with other anomaly detection techniques.

3.1 Models

To formulate our model, let us consider a dataset ($\mathbb{X}$) partitioned into a set of equally partitioned blocks ($D_{i,t}$), where i and t denote

block index and data point index within a block, respectively. Assuming p is the total number of partitioned blocks, and N is the total number of data points in each block, each data point can be represented as $D_{i,t}$, where $1 \leq i \leq p$ and $1 \leq t \leq N$. We also define a parameter, $\theta(D_i)$, which characterizes whether the block under consideration is associated with data points exhibiting typical or anomalous behavior. Let $\theta^{(i)}$ denote the parameter associated with the behavior or pattern of n data points in D_i (i.e., within a block) to determine whether a block contains anomalous data. Each block contains data points, $D_{i,t}$, that follow the condition, as in Equation 1.

$$D = \begin{Bmatrix} D_{1,t}, \text{ if } s_1 \leq t \leq e_1, [\theta^{(1)}] \\ D_{2,t}, \text{ if } s_2 \leq t \leq e_2, [\theta^{(2)}] \\ \cdots \\ D_{p,t}, \text{ if } s_p \leq t \leq e_p, [\theta^{(p)}] \end{Bmatrix}, 1 \leq i \leq p. \quad (1)$$

Next, we define the sparse representation, denoted as SR_i and AD_i for approximating data points for each partitioned block using Equation 2 below, which represent the characteristics of data points in each block. SR_i and AD_i are similar, except that AD_i includes additional information required for data reconstruction. Therefore, there is a correlation between $\theta(D_i)$ and the probability of anomaly behavior in block i as in Equation 3 because anomaly data has high SR_i and $AD(D_{i,t})$ values.

$$AD_i(D_{i,(s_i:e_i)}) \approx SR_i, 1 \leq i \leq p. \quad (2)$$

$$AD_i(D_{i,(s_i:e_i)}, \theta^{(i)}) = SR_i \times AD(D_{i,t}). \quad (3)$$

We consider two types of anomalies. For point anomaly, we calculate the number of anomaly points m according to the corruption rate α. To inject anomaly data points, we randomly select n data points out of N data points in data block D_i. Next, we replace the $D_{i,t}$ $(1 \leq t \leq N)$ with the generated n anomalous data points into D_i. For collective anomaly, we randomly choose the start of anomalous data points, denoted as t_s. Next, the end of anomaly points, t_{s+n} is defined by n according to the contamination rate α. Finally, from D_{i,t_s} to $D_{i,t_{s+n}}$ are replaced with the generated anomalous data points.

3.2 Characterizing Sparse Representations

Our approach begins by approximating raw data to extract SR_i, which compacts sparse representation for each block. To illustrate how this mechanism works, let us consider X, which denotes the original dataset in the transformed domain (DCT in our case). Note that the sum of information stored using the entire transformed components is 100%-percentile. We then select the K-dominant components ($\{X_1, X_2, ...X_k\}$) from the transformed coefficients as sparsity profiles.

While our approximation mechanism reduces data requirements by maintaining K transform components only (and indices associated with them for a reconstruction), selecting the right K, especially systematically, is not easy. One of the main reasons is that K is data-dependent, i.e., data in different applications exhibit different sparsity. In other words, K values vary spatially (e.g., block by block) and temporally (e.g., between checkpoints). In this paper, we propose two mechanisms to obtain K.

The first mechanism derives K using a predefined fixed information compaction rate. In this scheme, we acquire the full sample signal $X(t)$ (in the transformed domain), sort $|X(t)|$ in descending order, and determine the K largest (or dominant) components, defined by N-percentile information. For example, Fixed-95 uses K values that amount to 95%-percentile. We formulate this as:

$$E(X_k) = \frac{\sum_{i=1}^{k} X_i^2}{\sum_{i=1}^{n} X_i^2}, N = 1, 2, ..., n, k \leq n. \quad (4)$$

The second mechanism determines K using the Kneedle algorithm [32], denoted as Kneedle. In Kneedle, rather than using a fixed $E(X_k)$, we accomplish finding SR_i by detecting the 'knee-point' as described in [32]. The mathematical definition of 'knee' is given as a function of its first and second derivatives, formulated as:

$$K_f(x) = \frac{f''(x)}{(1 + f'(x)^2)^{1.5}}, \quad (5)$$

where $K_f(x)$ defines the curvature of f.

To use the Kneedle algorithm in the context of this paper, we fit the CDF of transformed coefficients into a smoothing spline to preserve the distribution's overall behavior, like two solid curves in Figure 3. Then we normalize the points, as in the x-axis in Figure 3, and find the knee points (K) of the normalized curve, typically the point of maximum curvature of the normalized curve shown in Figure 3. In other words, the knee point is the local maxima, depicting the maximum distance between the normalized curve and the line $y = x$.

Figure 3 shows the use of our knee detection algorithm on two selected blocks in one of our evaluated datasets: normal (blue) and abnormal (red). The dotted line curves depict the difference between the normalized spline (solid line curves) and $y = x$. The dashed vertical lines indicate the position of local maxima (maximum curvature of the normalized curve, i.e., the maximum distance between the spline and difference curve). These curves demonstrate that the block with abnormal vs. normal data points shows a noticeable difference, 0.1 vs. 0.3, in the normalized number of coefficients (i.e., the x-axis). Note that the x-axis in Figure 3 is normalized to make the detected knee points between 0 and 1.

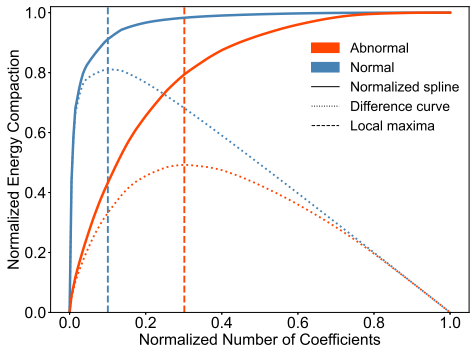


Figure 3: An illustration of finding anomaly using knee detection algorithm.

3.3 Anomaly Detection Algorithm

As we have presented, our anomaly detection mechanism utilizes the nature of the transformation we employed. We first find SR_i for each block i and the boundary threshold (τ) between *normal* and *abnormal*. Our detection mechanism determines τ using Max(SR_i) and Avg(SR_i) obtained during the training step for data blocks

Table 1: The Anomaly Detection Rule for all datasets.

Step	Data Blocks	→	Sparse Representation for Blocks	→	Characteristics of Anomaly Detection
Training	$D_1, D_2..D_{i-1}$	→	$SR_1, SR_2, ..., SR_{i-1}$	→	K_{max} = Max(SR_p), K_{avg} = Avg(SR_p), $p = D_1, D_2, \ldots, D_{i-1}$
Detection	$D_i, D_{i+1}, ...$	→	SR_i	→	Anomaly Boundary $\tau = K_{max} + K_{avg} \times w_i, SR_{B_i}$

Table 2: Evaluated Datasets and their characteristics.

Code	Datasets	Description	Size	KPSS test (p-value)	Stationarity
CMIP5	rlds	Surface downwelling longwave radiation	218MB	0.01	non-stationary
	mrsos	Moisture content of soil layer	218MB	0.01	non-stationary
FLASH	Sedov	Hydrodynamical test code involving strong shocks and non-planar symmetry	576MB	0.08	stationary
	Cellular	Burn simulation: cellular nuclear burning problem	1.35GB	0.01	non-stationary
Nek5000	Eddy	2D solution	820MB	0.1	stationary
	Vortex	Inviscid vortex propagation	580MB	0.01	non-stationary

$D_1, ..., D_p$, as defined in Table 1. As we have shown, K is notably different in blocks with anomalies. In other words, the higher the deviation from the original data points, the less compaction (more dispersed) in the transformed domain. Therefore, the determined τ is located between two distinct SR values, e.g., between 0.1 and 0.3 in the normalized coefficient domain, as shown in Figure 3. For instance, our algorithm chose τ of 24 (Kneedle) and 36 (Fixed-95) according to Table 1 in the detection step for the Eddy dataset.

4 EVALUATIONS

4.1 Setup

4.1.1 Datasets. We use six real-world scientific datasets from three production-level codes: FLASH, CMIP5, and Nek5000, as presented in Table 2. We use Kwiatkowski-Phillips-Schmidt-Shin (KPSS) to check the stationarity, which means that the statistical properties in a partitioned block, i.e., mean, variance, and covariance, do not change within a block. As the results show, Eddy and Sedov are relatively stationary, where the p-value of KPSS is lower than 0.05 (significance level).

4.1.2 Generating the Training and Testing Datasets. We use 30% of the data for training, which contains normal data points only, and the remaining 70% for testing. In the training datasets, we use the corruption rate (α) of 1%, 5%, and 10% to evaluate the performance of the evaluated schemes. We evaluate the corruption of 10% or lower because beyond that is rare in real scenarios and, more importantly, makes it hard to differentiate between normal and abnormal conditions. According to the corruption rate, we calculate the number of anomalous points n. Let i be the anomaly blocks in testing blocks (70% of the datasets) and N be the number of data points in each block. We replace n normal $D_{i,t}$ with n generated anomalous $D'_{i,t}$ in the 20% of randomly selected blocks from the testing blocks. For example, if α is 1%, 1% of the data points in each anomaly block are selected as anomalies. Then, the actual number of anomaly data points (or anomalies) is $N \times 0.7 \times 0.2 \times 0.01$, where N is the total number of data points.

4.1.3 Evaluated Schemes and Metrics. We compare our approaches (Fixed-95 and Kneedle) with two state-of-the-art unsupervised anomaly detection methods [18]: LoF [8] and AE [1, 25]. To evaluate the performance of the evaluated anomaly detection techniques, we implement an anomaly data injection model, consisting of point anomalies and collective anomalies using the library https://github.com/KDD-OpenSource/agots. We use the following metrics to assess the overall anomaly detection rates and how accurately the obtained SR approximates the original data. The latter ultimately dictates how closely the approximated data captures the original data.

- Accuracy (ACC) and False Positive Rate (FPR) are calculated from the ML confusion matrix.
- Approximation Ratio (AR) is given by: $AR = \frac{|D|-|D'|}{|D|} \times 100\%$, where $|D|$ is the size of D, $|D'|$ is the approximated data size. Error rate using PSNR (Peak Signal-to-Noise Ratio), $PSNR = 20 \cdot log10(\text{value range}) - 10 \cdot log10(MSE)$, where value range and MSE refer to the data value range and the mean squared error, respectively.

4.2 Results

4.2.1 Detection Accuracy. Figure 4 shows the detection accuracy of the evaluated schemes for point and collective anomalies while varying the ratio of data corruption α (1%, 5%, and 10%). Compared to AE and LoF, which perform relatively poorly in some datasets, our proposed schemes achieve high detection accuracy: 73.79%–98.26% for the point anomaly and 70.74%–100% for the collective anomaly. Experiments also demonstrate consistently lower FPR: 0% – 8.89% for point anomalies and 0% – 10.17% for collective anomalies.

Our approach also shows improving detection accuracy as the corruption rate increases. This trend is because, when the corruption rate gets higher (i.e., more anomalous data points in a block), higher K-dominants in an SR occur, leading to more apparent separation of τ from blocks with anomalous data points. AE and LoF, on the other hand, show deteriorating detection accuracy with lower contamination rates. We attribute this contrary trend to the fact that the AE-based anomaly detector uses the reconstruction error as an anomaly score. So when fewer anomalies exist, the reconstruction error between normal and abnormal blocks is not apparent. In other words, AE could become unstable in practical unsupervised cases without a reliable anomaly score, which is hard to determine automatically.

Similarly, LoF uses an isolated ratio as an anomaly score. In LoF, because the degree of anomaly depends on how isolated the object is from the surrounding neighborhood, it is also more challenging to decide the anomaly threshold for each block when there are fewer anomalies (or a lower corruption rate). Lastly, LoF and AE show slightly lower detection accuracy than ours for collective anomalies in some cases. We used the static anomaly score for LoF and AE in both point and collective anomalies. We adjusted the anomaly score for AE and LoF to achieve maximum performance.

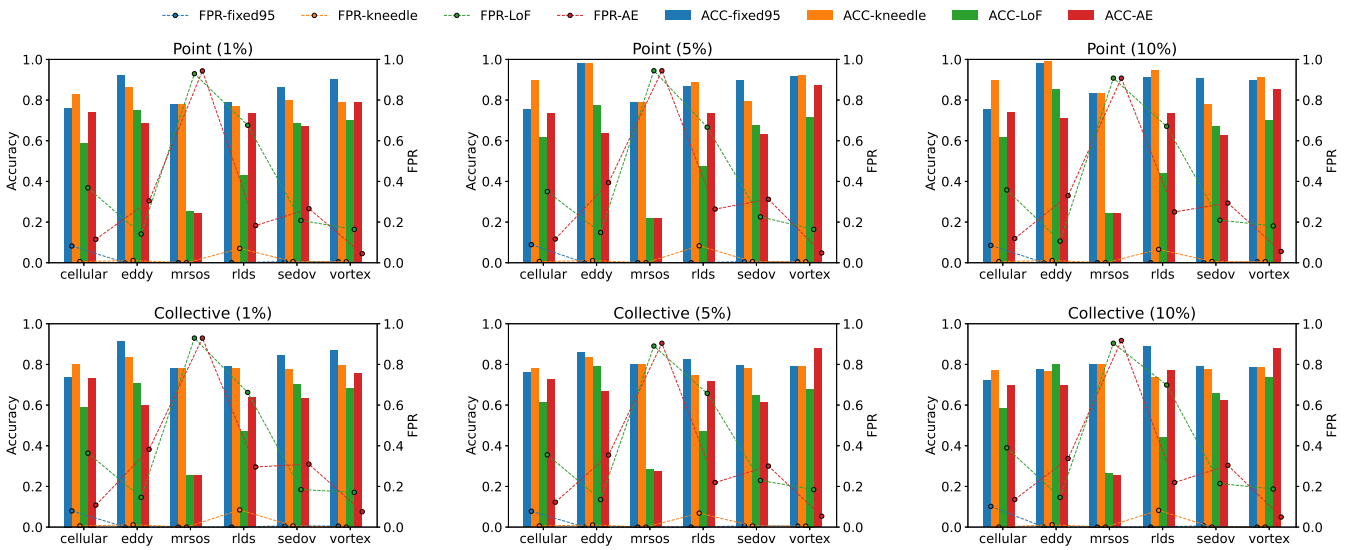


Figure 4: Comparison of detection accuracy (bar plot) and false positive rate (line plot).

Table 3: Comparison of approximation ratio for our approaches with different sparse representations.

	Cellular			Eddy			mrsos		
	AR	AS	PSNR	AR	AS	PSNR	AR	AS	PSNR
Fixed-95	99.7%	93.98%	22.86	85.04%	78.95%	27.83	76.76%	70.48%	22.95
Kneedle	90.16%	84.07%	28.94	87.41%	81.5%	29.97	84.99%	79.12%	22.25
	rlds			Sedov			Vortex		
	AR	AS	PSNR	AR	AS	PSNR	AR	AS	PSNR
Fixed-95	99.81%	94.17%	20.22	94.36%	88.49%	24.85	93.36%	87.44%	28.26
Kneedle	88.75%	82.59%	30.53	91.67%	85.86%	26.8	90.1%	84.03%	29.9

4.2.2 Approximation Performance. The high detection accuracy achieved by our schemes confirms our hypothesis that the relationship among the K-dominant coefficients allows effective anomaly detection in scientific datasets. To understand our sampling mechanism's efficiency in capturing the original data characteristics in a compact form, we measure the approximation ratio of fixed-information-based (Fixed-95) and Kneedle-based (Kneedle) because each finds an optimal sparse profile differently. Because AS means the actual data size to store K-dominant coefficients and their indices, our approach incurs the additional storage overhead computed by AR minus AS. Note that the approximation ratios presented in Table 3 are for all data blocks, including training and testing datasets. As we can see from Table 3, all schemes approximate data well with high ratios. For example, our approach can achieve up to 99.7% of ARs using Fixed-95 for Cellular, which means that the sparse profile of Cellular requires only 0.3% of the original data while achieving high detection accuracy.

We next evaluate the reconstructed data to measure how the quality of the approximated data is comparable to the original data, which led to higher detection accuracy. The PSNR column in Table 3 shows the error rates between the reconstructed and the original data.; in other words, the higher the PSNR, the smaller the error. In our proposed techniques (Fixed-95 and Kneedle), K varies depending on data, i.e., variable data approximation, and so does in PSNR values, as shown in Table 3. Overall, the PSNR of Fixed-95 generally is lower than that of the PSNR of Kneedle but with higher approximation ratios, e.g., in the case of Cellular, 28.86 of PSNR with 99.7% of AR and 28.94 of PSNR with 90.16% of AR for Fixed-95 and Kneedle, respectively. These results confirmed that our approach achieves a competitive approximation ratio while maintaining a relatively low error rate.

5 CONCLUSIONS AND FUTURE WORK

This paper proposes an unsupervised anomaly detection model for scientific datasets. The motivation behind our method is that scientific datasets exhibit a certain degree of sparsity. We use discrete transformations, specifically DCT, to reveal the signal's spatial behavior, leading to efficient sparse representation. Our technique also exploits the correlation between the number of dominant transform coefficients and data anomalies. To validate the proposed approach, we inject anomalies into original scientific simulation datasets and compare performance with the other two state-of-the-art unsupervised anomaly detection techniques, AE and LoF. Our experimental results show that our approach achieves higher anomaly detection accuracy with an increasing contamination rate. Our proposed schemes successfully detect anomalies while obtaining about 90.16% of the approximation ratio, i.e., requiring only 9.84% of original Cellular data. We also observe that the detection accuracy by our schemes could be 75.54%–92.13% of point anomalies, unlike 28.82%-34.49% in AE and 58.96%-61.57% in LoF. These results demonstrate that our proposed approach can achieve a high approximation/compression ratio while accurately detecting anomalies.

ACKNOWLEDGMENT

This material is in part based upon work supported by the National Science Foundation under Grant No. 1751143 and 2312982. The Titan X Pascal used for this research was donated by the NVIDIA Corporation. This work is also supported by the Korea Innovation Foundation (INNOPOLIS) grant funded by the Korean government (MSIT) (2020-DD-UP-0278).

REFERENCES

[1] Charu C Aggarwal. 2015. Outlier analysis . In *Data mining*. 75–79.

[2] R. Ahad, E. Chan, and A. Santos. 2015. Toward autonomic cloud: Automatic anomaly detection and resolution. In *International Conference on Cloud and Autonomic Computing, ICCAC*.

[3] Tsatsral Amarbayasgalan, Van Huy Pham, Nipon Theera-Umpon, and Keun Ho Ryu. 2020. Unsupervised Anomaly Detection Approach for Time-Series in Multi-Domains Using Deep Reconstruction Error. *Symmetry* (2020).

[4] Leonardo Bautista-Gomez and Franck Cappello. 2014. *Detecting Silent Data Corruption for Extreme-Scale Applications through Data Mining*. Technical Report ANL/MCS-TM-346. Argonne National Laboratory.

[5] Eduardo Berrocal, Leonardo Bautista-Gomez, Sheng Di, Zhiling Lan, and Franck Cappello. 2015. Lightweight Silent Data Corruption Detection Based on Runtime Data Analysis for HPC Applications. In *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing(HPDC)*.

[6] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. 2018. Anomaly Detection using Autoencoders in High Performance Computing Systems . In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[7] Andrea Borghesi, Antonio Libri, Luca Benini, and Andrea Bartolini. 2019. Online Anomaly Detection in HPC Systems . In *IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*.

[8] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 93–104. https://doi.org/10.1145/342009.335388

[9] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep Learning for Anomaly Detection: A Survey. *Journal of Big Data* (2019). https://arxiv.org/abs/1901.03407v2

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

[11] Jieyang Chen, Hongbo Li, Sihuan Li, Xin Liang, Panruo Wu, Dingwen Tao, Kaiming Ouyang, Yuanlai Liu, Kai Zhao, Qiang Guan, and Zizhong Chen. 2018. Fault Tolerant One-sided Matrix Decompositions on Heterogeneous Systems with GPUs. In *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. 854–865. https://doi.org/10.1109/SC.2018.00071

[12] Zizhong Chen. 2013. Online-ABFT: An Online Algorithm Based Fault Tolerance Scheme for Soft Error Detection in Iterative Methods. In *Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (Shenzhen, China) *(PPoPP '13)*. Association for Computing Machinery, New York, NY, USA, 167–176. https://doi.org/10.1145/2442516.2442533

[13] Mohamed Cherif Dani, Henri Doreau, and Samantha Alt. 2017. K-means Application for Anomaly Detection and Log Classification in HPC. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*.

[14] Sheng Di and Franck Cappello. 2016. Adaptive Impact-Driven Detection of Silent Data Corruption for HPC Applications. *IEEE Transactions on Parallel and Distributed Systems* (2016), 2809–2823.

[15] Paul Fischer, James Lottes, Stefan Kerkemeier, Oana Marin, Katherine Heisey, Aleks Obabko, Elia Merzari, and Yulia Peet. 2015. *Nek5000 User Documentation*. Technical Report ANL/MCS-TM-351. Argonne National Laboratory.

[16] Flash Center for Computational Science. 2016. FLASH User's Guide.

[17] Al Geist. 2016. Supercomputing's monster in the closet. *IEEE Spectrum* 53, 3 (2016), 30–35. https://doi.org/10.1109/MSPEC.2016.7420396

[18] Markus Goldstein and Seiichi Uchida. 2016. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data . In *PLoS ONE 11(4): e0152173. https://doi.org/10.1371/journal.pone.0152173*.

[19] M.A Hayes and M.A. Capretz. 2015. Contextual anomaly detection framework for big sensor data . *Journal of Big Data* (2015). https://doi.org/10.1186/s40537-014-0011-y

[20] H. Jayathilaka, C. Krintz, and R. Wolski. 2017. Performance monitoring and root cause analysis for cloud-hosted web applications. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*.

[21] Sihuan Li, Jianyu Huang, Ping Tak Peter Tang, Daya Khudia, Jongsoo Park, Harish Dattatraya Dixit, and Zizhong Chen. 2021. Efficient Soft-Error Detection for Low-precision Deep Learning Recommendation Models. https://doi.org/10.48550/ARXIV.2103.00130

[22] Xin Liang, Jieyang Chen, Dingwen Tao, Sihuan Li, Panruo Wu, Hongbo Li, Kaiming Ouyang, Yuanlai Liu, Fengguang Song, and Zizhong Chen. 2017. Correcting Soft Errors Online in Fast Fourier Transform. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, Colorado) *(SC '17)*. Association for Computing Machinery, New York, NY, USA, Article 30, 12 pages. https://doi.org/10.1145/3126908.3126915

[23] Luis Martí, Nayat Sanchez-Pi, José Manuel Molina, and Ana Cristina Bicharra Garcia. 2015. Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. *Sensors* (2015).

[24] Gerald A. Meehl, Curt Covey, Bryant McAvaney, Mojib Latif, and Ronald J. Stouffer. 2005. Overview of the Coupled Model Intercomparison Project. *Bulletin of the American Meteorological Society* 86, 1 (2005), 89–93.

[25] Nicholas Merrill and Azim Eskandarian. 2020. Modified Autoencoder Training and Scoring for Robust Unsupervised Anomaly Detection in Deep Learning. In *IEE Access*. 101824–101833.

[26] Aekyeung Moon, Jaeyoung Kim, Jialing Zhang, Hang Liu, and Seung Woo Son. 2017. Understanding the Impact of Lossy Compression on IoT Smart Farm Analytics. In *2017 IEEE Big Data*. 4602–4611.

[27] Aekyeung Moon, Xiaoyan Zhuo, Jialing Zhang, Seung Woo Son, and Yun Jeong Song. 2020. Anomaly Detection in Edge Nodes using Sparsity Profile. In *Proceedings of IEEE Big Data*. 1236–1245.

[28] Bin Nie, Ji Xue, Saurabh Gupta, Christian Engelmann, Evgenia Smirni, and Devesh Tiwari. 2017. Characterizing Temperature, Power, and Soft-Error Behaviors in Data Center Systems: Insights, Challenges, and Opportunities. In *2017 IEEE 25th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 22–31. https://doi.org/10.1109/MASCOTS.2017.12

[29] Sean Peisert, George Cybenko, Sushil Jajodia, David L. Brown, Christopher L. DeMarco, Paul Hovland, Sven Leyffer, Celeste Matarazzo, Stacy Prowell, Brian Tierney, and Von Welch. 2015. ASCR Cybersecurity for Scientific Computing Integrity. Department of Energy Office of Science.

[30] Sean Peisert, Thomas E. Potok, and Todd Jones. 2015. ASCR Cybersecurity for Scientific Computing Integrity – Research Pathways and Ideas Workshop. Departmet of Energy Office of Science.

[31] Hansheng Ren, Bixiong Xu, Chao Yi Yujing Wang, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft . In *KDD*.

[32] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. 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 (ICDCSW '11)*. IEEE Computer Society, USA, 166–171. https://doi.org/10.1109/ICDCSW.2011.20

[33] Omer Subasi, Sheng Di, Leonardo Bautista-Gomez, Prasanna Balaprakash, Osman Unsal, Jesus Labarta, Adrian Cristal, and Franck Cappello. 2016. Spatial Support Vector Regression to Detect Silent Errors in the Exascale Era. In *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. 413–424. https://doi.org/10.1109/CCGrid.2016.33

[34] Omer Subasi, Sheng Di, Leonardo Bautista-Gomez, Prasanna Balaprakash, Osman Unsal, Jesus Labarta, Adrian Cristal, Sriram Krishnamoorthy, and Franck Cappello. 2018. Exploring the capabilities of support vector machines in detecting silent data corruptions. *Sustainable Computing: Informatics and Systems* 19 (2018), 277–290. https://doi.org/10.1016/j.suscom.2018.01.004

[35] Dingwen Tao, Shuaiwen Leon Song, Sriram Krishnamoorthy, Panruo Wu, Xin Liang, Eddy Z. Zhang, Darren Kerbyson, and Zizhong Chen. 2016. New-Sum: A Novel Online ABFT Scheme For General Iterative Methods. In *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing* (Kyoto, Japan) *(HPDC '16)*. Association for Computing Machinery, New York, NY, USA, 43–55. https://doi.org/10.1145/2907294.2907306

[36] Ozan Tuncer, Emre Ates, Yijia Zhang, Ata Turk, Jim Brandt, Vitus J. Leung, Manuel Egele, and Ayse K. Coskun. 2017. Diagnosing Performance Variations in HPC Applications Using Machine Learning. In *International supercomputing Conference, Springer*.

[37] Ozan Tuncer, Emre Ates, Yijia Zhang, Ata Turk, Jim Brandt, Vitus J. Leung, Manuel Egele, and Ayse K. Coskun. 2019. Online Diagnosis of Performance Variation in HPC Systems Using Machine Learning. In *IEEE transactions on parallel and distributed systems*.

[38] Chen Wang, Nikoli Dryden, Franck Cappello, and Marc Snir. 2018. Neural Network Based Silent Error Detector. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. 168–178.

[39] Panruo Wu, Nathan DeBardeleben, Qiang Guan, Sean Blanchard, Jieyang Chen, Dingwen Tao, Xin Liang, Kaiming Ouyang, and Zizhong Chen. 2017. Silent Data Corruption Resilient Two-Sided Matrix Factorizations. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (Austin, Texas, USA) *(PPoPP '17)*. Association for Computing Machinery, New York, NY, USA, 415–427. https://doi.org/10.1145/3018743.3018750

[40] Jialing Zhang, Aekyeung Moon, Xiaoyan Zhuo, and Seung Woo Son. 2019. Towards Improving Rate-Distortion Performance of Transform-Based Lossy Compression for HPC Datasets . In *IEEE HPEC*.

[41] Jeff Zhang, Kartheek Rangineni, Zahra Ghodsi, and Siddharth Garg. 2018. ThUnderVolt: Enabling Aggressive Voltage Underscaling and Timing Error Resilience for Energy Efficient Deep Learning Accelerators. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1109/DAC.2018.8465918

[42] Jialing Zhang, Xiaoyan Zhuo, Aekyeung Moon, Hang Liu, and Seung Woo Son. 2019. Efficient Encoding and Reconstruction of HPC Datasets for Checkpoint/Restart. In *Symposium on Mass Storage Systems and Technologies (MSST)*. https://doi.org/10.1109/MSST.2019.00-14