Fault Prediction in Planetary Drilling Using Subspace Analysis Techniques

Sarah Boelter*[1 [0009−0004−2298−1310]], Lucas Weber*[2 [0000−0002−6877−6935]], Richard Lenz[2 [0000−0003−1551−4824]], Brian Glass[3 [0000−0002−3559−5249]], and Maria Gini[1 [0000−0001−8841−1055]]

[1] University of Minnesota Twin-Cities, Minneapolis, Minnesota, 55416
{boelt072, gini}@umn.edu
[2] Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany, 91058
{lucas.weber, richard.lenz}@fau.de
[3] NASA Ames Research Center, Moffett Field, California, 94043
brian.glass@nasa.gov

Abstract. In remote planetary environments, robotic agents must respond to or reason through faults before they escalate to mission-critical failures. No broadly applicable solution exists to give a specialized agent like The Regolith and Ice Drill for Exploring New Terrain (TRIDENT) situational awareness for when a situation may escalate to a drilling fault. We propose a new online time-series subspace analysis method, Entangled Singular Spectrum Transformation (ESST), to better predict and analyze faults using online data produced by the TRIDENT drill. We evaluate performance against other online subspace analysis techniques to determine the optimal detection method for sudden drilling faults.

Keywords: singular spectrum analysis · fault detection · time series

1 Introduction

In 2024, NASA Space Technology Mission Directorate (STMD) released feedback on 187 shortfalls to better understand civil space's most pressing technology shortfalls [21]. There is need for robotic autonomy for mitigating operational risk in challenging environmental conditions, often with hardware and software limitations. This paper focuses on a robotic drill, known as "The Regolith and Ice Drill for Exploring New Terrain" (TRIDENT), a 1-meter rotary percussive drill manufactured by Honeybee Robotics designed to generate cuttings from a borehole for further scientific analysis. We spent time field testing the TRIDENT drill with NASA Ames in the Bishop Tuff Fall 2023 and at a Haughton Crater Analog site in 2024. It was observed TRIDENT gives indicators of a fault before it happens. Manually analyzed time series data confirms this, but necessary components to intelligently monitor the drill have not been developed. Due to the difficulty of collecting data and the sparsity of fault datasets, training a

* These authors contributed equally to this work.

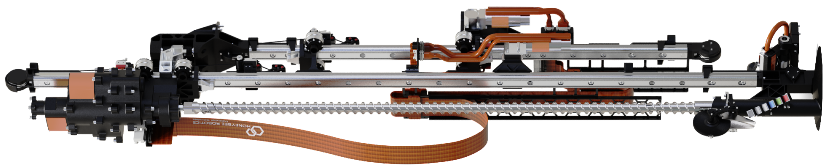

Fig. 1. Trident Drill [1]

supervised model becomes challenging. This necessitates developing models of fault detection that accurately can predict faults online with little to no training data.

We propose predicting situational risk factors for critical errors for TRIDENT using change point analysis techniques building off previous work [4], which will aid in drill fault avoidance. We propose a new time-series change analysis technique, the Entangled Singular Spectrum Transformation (ESST), which builds upon the Singular Spectrum Transformation (SST) [13]. Instead of computing the decomposition and comparing the extracted characteristics separately, ESST combines both trajectory matrices and computes the score directly after decomposition using more characteristics than the SST. An empirical comparison of other algorithms with ESST shows how our method works for our scenario.

2 Project Background

The Atacama Rover Astrobiology Drilling Studies (ARADS) project aimed to explore and understand the mobility and distribution of salts, compounds, biosignatures, and extant life down to a 1 meter depth in planetary environments using a rover with an attached drill. This system was tested in Chile in September 2019. Diagnostic and automation software was prototyped [11].

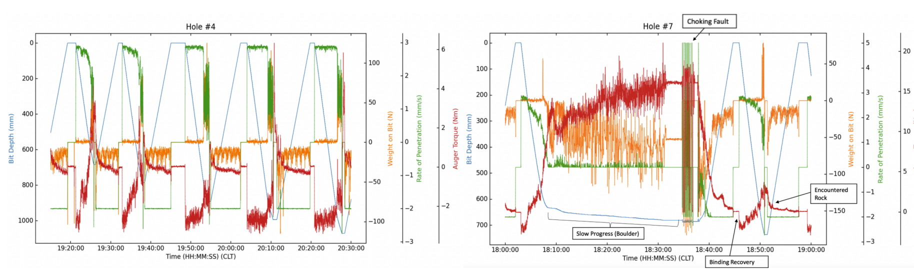

Fig. 2. The left image shows normal drilling operations during testing. The right image illustrates choking fault during testing. Both testing done in Devon Island, Canada [12]

It is impossible to know without prior geological surveying what rock types exist below the surface and what will be encountered. It will be difficult to have a one-size-fits-all program to all scenarios, since drilling techniques differ with changes in target composition, in addition to external factors like temperature.

We collected 30 instances of faults in 2024 and two in 2023 from Haughton Crater at Devon Island, Canada and at the Goddard Instrument Field Team's September deployment to the tuff deposits near Bishop, CA [12]. Figure 2 shows time-series examples from two boreholes. The first shows normal drilling operation and the second illustrates a fault. From observing similar plots, we inferred faults tend to raise variables like Torque (force applied by motor) and Weight On Bit (force applied to the bit surface area by drill string) while the Rate of Penetration (velocity of drill string relative to y axis) is roughly zero. We need methods able to detect shifts under 30 seconds in sparse amounts of data.

3 Related Work

3.1 Autonomy in Petroleum and Mining

We must consider related fields and how they have addressed drilling fault-avoidance. Recently, NORCE, the Norwegian Research Center, conducted research into autonomous drilling in petroleum. Specifically, de Wardt et al. [23] published a taxonomy for autonomous drilling. Due to complexity, uncertainty, and sparse observability inherent in drilling, automation technologies applicable to other industries like aviation may not be applicable to drilling [23].

A complication inherent to petroleum drilling relevant to planetary drilling is weak observability of the environment. The uniqueness of sparse data observed from a distance makes autonomy challenging compared to other industries. To estimate the subsurface environment and drilling operations state, questions about the drilling operation must be asked, which are often difficult to answer accurately [23]. To achieve autonomous operations, methods must estimate the internal state of the drilling system and make decisions. Cayeux et al. [6] developed a method utilizing a Markov Decision Process to optimize operations to quickly react to abnormal drilling while minimizing time lost to unexpected drilling events. Some issues taken into account by Cayeux et al. are not relevant to our problem, specifically the flow rate and friction tests. In addition actions in planetary drilling are much simpler and less repetitive.

Meanwhile, the Australian Centre for Field Robotics is focused on automation in the surface mining industry to improve safety and efficiency in open-pit mining operations in the Pilbara. Leung et al. [15] discuss robotic sensing and automation development for blasthole drilling, where a hole is drilled for an explosive to be dropped and charged. In addition to autonomous drill development, a solution is currently under development is a down hole inspection robot (DHIR) to overcome the lack of feedback between the planning and drilling stage, and study the structural integrity of holes over time between drilling and charging.

3.2 Change Point Detection and Signal Segmentation

Time series Change Point Detection (CPD) and segmentation are related topics, with a broad range of methods and applications [22, 3, 9, 24]. Methods aim to find discrete points where signals change to identify when the generating process has changed. A clear definition depends on the respective application areas. For our use case, we identified the following requirements:

1. Online Detection: To process and detect incoming data online to catch fault indications.
2. Unsupervised: Datasets are few and difficult to obtain, therefore, training of supervised models is out of scope.
3. Data Efficiency: Drilling is unobservable and unpredictable, therefore methods needing little calibration are preferred.
4. Computational Efficiency: Because of limited computational resources, we require the algorithms to have low computational costs.
5. Robust: Data is noisy due to percussion and other effects, necessitating a parameter-free method robust to these circumstances.

Since we require online detection and unsupervised algorithms, we rule out methods such as BinSeg and PELT, and the large field of supervised approaches [3]. Due to the third and fourth requirements, we exclude algorithms using reconstruction-based methods, often employing auto-encoders like TIRE [7]. These neural network approaches require large computational resources and longer signals for their unsupervised training. Additionally, they might require retraining once the drilling conditions change. Statistical methods assume the points in a time series originate from a statistical process and aim to detect when that process changes.

Commonly used algorithms are Bayesion Online Change Point Detection (BOCPD) [2] and Relative unconstrained Least-Squares Importance Fitting (RuLSIF) [16]. BOCPD uses Bayesian statistics and conjugated priors to derive a run-length probability of unvaried signal segments. RulSIF estimates and compares the probability distribution before and after point t in a time series. We use RuLSIF as the best-performing method.

From the field of time series segmentation, we choose Fast Low-Cost Online Segmentation (FLOSS) [10] and Classification Score Profile (ClaSP) [19, 9] as the most recent and best-performing algorithms. FLOSS finds change points by comparing how many nearest neighbor relations cross a certain point t in a time series using the matrix profile. ClaSP builds on the idea that if a change occurred at time t, a binary classifier will be able to distinguish between before and after t in a binary classification scenario. While the proposed algorithm is not online, the idea can be trivially extended using a classifier within a sliding window. Subspace methods are another way of detecting changes in the shape characteristics of time series. Building on the extensive theory of Singular Spectrum Analysis (SSA), these methods use time series decomposition to extract characteristic subsequences before and after some point t and compare their shape to detect changes. Different methods mainly differ in how the characteristics are extracted and utilized for comparison [18, 13, 14, 17].

4 Methods

4.1 An Argument for Subspace-based Algorithms

In the last section, we derived requirements and applied them to the field of CPD to select approaches for our use case. While we evaluate other methods, we want to provide a reason for focusing on subspace-based methods. Statistical methods work well for distributional changes of probabilistic systems, but this is not our goal when identifying drilling faults. Regarding FLOSS and ClaSP, we expect

Table 1. Overview of notations used in Section 4.

Symbol	Meaning
$T_{i:j}$	A subsequence of the time series T. Subsequences are continuous and contain all samples from index i to and including j.
$w,\ n \in \mathbb{N}$	The window length w and number of windows n used for the change point scoring.
$H_\mathrm{p} \in \mathbb{R}^{w \times n}$	H is the trajectory matrix constructed for the decomposition. The matrix has Hankel structure and contains samples from $T_{p:(p+w+n-1)}$.
S_T	The scoring sequence with the same length as T. S_T^i: score at time i.
$A^i \in \mathbb{R}^{w \times 1}$	The column of arbitrary matrix A. For example, H_p^i contains the subsequence $T_{(p+i):(p+w+i-1)}$
$A^{i,j}$	The j-th element in the i-th column of arbitrary matrix A.
$(A)^T$	Superscript T denotes the transpose of arbitrary matrix A.
$U_{p,k} \in \mathbb{R}^{w \times k}$	The left singular vectors after computing the decomposition of matrix H_p and retaining only the k most important singular vectors.
$\Sigma_{p,k} \in \mathbb{R}^{1 \times k}$	The k most important singular values are sorted in decreasing order on the main diagonal of an otherwise zero-valued matrix.
$V_{p,k} \in \mathbb{R}^{k \times w}$	The left singular vectors (similar to $U_{p,k}$). $\mathrm{SVD}_k(H_p) = U_{p,k}\Sigma_{p,k}(V_{p,k})^T$.

data efficiency and detection delay could become a problem. Both algorithms need relatively many subsequences to properly find nearest neighbors before t (data efficiency) and after t (detection delay). This could become a problem, as relatively short drilling cycles might not be enough to build an appropriate set of comparable sequences. Additionally, the percussive noise will affect the Euclidean distances both methods rely on.

Subspace-based methods are best at detecting instantaneous changes while not requiring an extensive collection of historical sequences. They also do not extract statistical properties but mainly characteristic shapes and ignore noise, as the decomposition acts as a filter for random signal contributions. While noise filtering can be a disadvantage for statistical CPD [16], and the lack of a "long-term memory" reduces their usability for segmentation, we think that these methods are promising for the unique properties of our application scenario.

4.2 Singular Spectrum Transformation

The SST [13, 14] is the most commonly used subspace method. It detects changes by computing and comparing subspaces extracted from the original time series using time series decomposition. This section will introduce the necessary foundation to understand the original SST, which we will then build upon to introduce ESST in the next section. See Table 1 for an overview of the notation.

Subspace-based CPD methods use the Singular Value Decomposition (SVD) of two trajectory matrices H_i and H_j with $i < t < j$ to extract the representative sequences before and after $(+\delta t)$ some point in time t and compare them to provide a change point score. Mathematically, the score for the SST S_T^t for time series T and time t is computed using the following equation [14]:

$$S_T^t = 1 - ||(U_{t,k})^T U_{t+\delta t}^1||^2 = 1 - \sum_{i=0}^{k-1} (\underbrace{(U_t^i)^T U_{t+\delta t}^1}_{\text{Vector Product}})^2 \tag{1}$$

where $U_{t,k}$ denotes the most prominent k left singular vectors of the corresponding trajectory matrix H_t.

For an intuitive understanding, we can approach the SST from the viewpoint of linear algebra. The SVD decomposes a matrix into a collection of left singular vectors U, right singular vectors V, and singular values Σ that reconstruct the original matrix. When only considering first k vectors and values, the reconstruction is approximate but the best reconstruction possible in terms of Frobenius norm and ℓ_2 matrix norm using only this set of k vectors (a subspace) [8]. We can interpret the left singular vectors U_t as best representing all columns of the original matrix. In other words, if we use only one vector to represent all columns of the original matrix, the first left singular vector is the most appropriate one. The singular value is how much this left singular vector is contained in all columns of H_t overall. For example, if the first singular value is significantly larger than all others, the matrix overall can be well represented by the first singular vector (low rank). When trying to understand the right singular vectors, it helps to restructure the original SVD equation:

$$A \overset{\text{SVD}}{=} U_k \Sigma_k (V_k)^T \text{ with } A \in \mathbb{R}^{w \times n};\ U_k, (V_k)^T \in \mathbb{R}^{w \times k};\ \Sigma_k \in \mathbb{R}^{k \times k} \tag{2}$$

$$(U_k)^T A = \Sigma_k (V_k)^T \xrightarrow{k=1} [\underbrace{(U^0)^T A^0}_{\text{Vector Product}}, \ldots, (U^0)^T A^n] = \sigma_0 (V^0)^T \tag{3}$$

Eq. 3 shows the right singular vector is a collection of vector products of the left singular vector with the columns of the original matrix. Therefore, the right singular is interpreted as the similarity of each original matrix column with the first left singular vector. The j-th element $V^{i,j}$ of V^i is the similarity of U^i to the j-th column $A^j \in \mathbb{R}^{w \times 1}$ of the original matrix A.

Transferring all of the above to CPD, where the columns of the trajectory matrix H_t are time series subsequences, the left singular vectors U_t are also time series. With that, the first left singular U_t^0 is the time series that best represents all the time series contained in the original trajectory matrix H_t if one has to choose only one time series. The elements of the first right singular vector $V_t^{0,j}$ indicate how well U_t^0 represents each individual subsequence $H_t^j = T_{(t+j):(t+w+j-1)}$ of the original trajectory matrix H_t, and the singular value is the strength of representation over all columns of H_t. The SST score S_T^t boils down to comparing the k characteristic sequences $U_{t,k}$ from before t with the most characteristic sequence $U^0_{t+\delta t}$ after t. If they are similar, their vector product becomes one, and the score S^t_T is minimized, indicating no change.

The SST has several parameters to set. There is the rank k of the subspace, essentially how many representative sequences one wants to compare with. Then there is the number of sequences n and length w of the subsequences that make up the trajectory matrix. There is also the distance δt between the two trajectory matrices one wants to compare. While all of these parameters are necessary, the original authors recommend only setting w and keeping $n = w$ equal to that. They also set $\delta t = w/3$ to a third of the window size and fix $k = 5$ as appropriately sampled time series are expected to be low rank.

4.3 ESST - Modifying the Singular Spectrum Transformation

As can be seen in Eq. 1, the SST takes several characteristic sequences from the past trajectory matrix but only compares them to the most prominent one from the future trajectory matrix $H_{t+\delta t}$. While Idé [14] provides an argument for this, we have reasons to believe incorporating more information from $H_{t+\delta t}$ will improve the detections of our noisy sensor signals. Intuitively, large noise results in a slowly decreasing singular spectrum, meaning that the trajectory matrix cannot be represented well using only one singular vector. Consequently, an improved change score uses several vectors in addition to the most important singular vector from the future trajectory matrix. The related work supports using more information from $H_{t+\delta t}$. Moskvina and Zhigljavsky [18] compare all distances from the past characteristics with the future subsequences. Mohammad and Nishida [17] empirically demonstrate that using multiple future singular vectors for score computation improves the overall results.

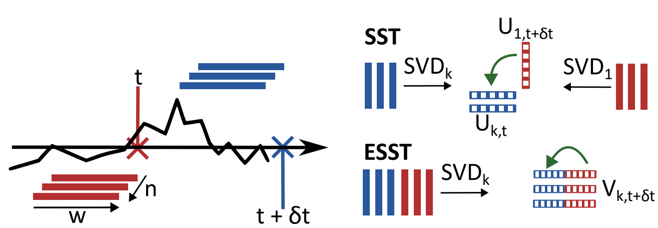

Fig. 4. Visualization of the difference between SST and ESST. The score computation is in green.

This can be done using metrics like pairwise distances or subspace angles [18, 17], but we argue that there is an even simpler way of incorporating more information from both trajectory matrices by simply using the right singular vectors that are a byproduct of the decomposition but have been mostly ignored.

As explained earlier, the right singular vectors express how much the corresponding left singular vector is contained in each column of the original trajectory matrix. If there is a significant variation between the elements in a right singular vector V^i, the corresponding left singular vector U^i does not represent all columns of the original matrix equally well.

Suppose we fill a trajectory matrix in a way where the first $n/2$ columns are subsequences before t and the second half are subsequences after t. If we then compare the values of the first $n/2$ elements $V^{i,0:\frac{n}{2}-1}$ with the second half of a right singular vector $V^{i,\frac{n}{2}:n-1}$, we can see whether there is a significant difference in how well the left singular vector represents the time series before and after t.

In other words, if we find that there are characteristic sequences in the first k right singular vectors that only represent the left or right side of our constructed trajectory matrix well but not the other half, our method detects a change. Quantitatively, we compute the change score by comparing the mean of the first $n/2$ values $V^{i,0:\frac{n}{2}-1}$ with the mean of the second half $V^{i,\frac{n}{2}:n-1}$ for each of the k first right singular vectors. We then fuse these scores by weighing them with their corresponding singular values (Eq. 4).

$$\tilde{S}_T^t = \frac{\sum_{i=0}^{k-1} \sigma_i \left| \frac{1}{N} \left(\sum_{j=0}^{n/2-1} V^{i,j} - \sum_{j=n/2}^{n-1} V^{i,j} \right) \right|}{\sum_{i=0}^{k-1} \sigma_i} \tag{4}$$

We name this modified method **Entangled SST** as we do not decompose the trajectory matrices separately but entangle both within a singular matrix. Regarding hyperparameter, we propose to mainly set w, deriving the other hyperparameters similarly to the SST. If the time series structure changes rapidly, we recommend setting n smaller than w. Fig. 4 compares SST and ESST.

4.4 Evaluation

The evaluation of different CPD algorithms is neither standardized nor generalizable. With the large heterogeneity in backgrounds of those developing the methods, there are equally as many methods used for evaluation. We selected the methods targeting the unique constraints of our application and evaluate methods for our use case. We mention this explicitly, as it is important for interpretation of the results. Just because methods do not perform well in our scenario does not immediately render them useless in other applications. As obtaining data is costly and faults are rare, we must restrict our evaluation to cases available to us. When analyzing the related work, we found change point detection is treated as a binary classification problem, where the decision is either change or no change [3, 5, 22]. While the related work often evaluates the measures on a collection of discrete change points, we evaluate the measurements on a per-sample basis in our use case with limited examples. As a consequence, we are dealing with a very skewed distribution, where most of the samples will have the class no change, and only a small minority will have the label change. Therefore, labeling all samples with no change would achieve high accuracy. To mediate this effect, we focused on the positive minority class by using the F1-score, which is the geometric mean of precision and recall, both of which mainly focus on the positive class (is change in our case). Our evaluation is comparable to the evaluation procedure for time series anomaly detection [20]. A problem, where the sparsity of positive data samples is similar to ours. In the time series anomaly detection field, there has been valid criticism of the evaluation benchmarks of many approaches [25]. To mitigate the effects of potentially mislabeled or misleading results due to the compression of the results into scalar metrics, we provide visual results showing how different methods score. As a baseline algorithm and sanity check if our problem is trivially solvable, we also implemented a baseline algorithm termed MEANVAR. It has two sliding windows, left and right of t, where we keep track of the mean and variance of the samples contained in each window. For a change score, we compute the difference in mean and variance of both windows and add them. For the sake of reproducibility, we will publish our evaluation code in the accompanying Github repository [4]. All CPD methods are available [5].

5 Results

The F1-Score, shown in Table 2,is a geometric mean of precision and recall scores. The closer a score is to 1, the higher the precision and recall. We desire scores above .6. In Table 2 we underline the best result and mark the second best with an asterisk. Numbers are rounded to two significant digits. ESST performed most consistently overall, with the majority of scores over .7. For the two ESST scores below .6, the scores were better than for other methods on the same dataset. MeanVar performed slightly better that ESST on a single dataset.

Table 2. Evaluation Results for the different CPD methods in terms of F1-Score. Best results are underlined and second best are asterisked. Higher is better.

Data	MeanVar	RuLSIF	FLOSS	ClaSP	SST	ESST
Torque1	0.11	0.23	0.01	0.06	0.48*	<u>0.85</u>
Torque2	<u>0.63</u>	0.46	0.25	0.10	0.36	0.56*
Torque3	0.71	0.34	0.49	0.09	0.63*	<u>0.76</u>
Torque4	0.61	0.63	0.64*	0.19	0.00	<u>0.83</u>
Torque5	0.19	0.00	0.00	0.09	0.36*	<u>0.39</u>
WOB1	0.35	0.22	0.38*	0.00	0.27	<u>0.89</u>
WOB2	0.32	0.11	0.48*	0.00	0.12	<u>0.71</u>
WOB3	0.34	0.17	0.40*	0.00	0.25	<u>0.84</u>
WOB4	0.80*	0.38	0.00	0.05	0.62	<u>0.92</u>
WOB5	0.33*	0.15	0.00	0.00	0.15	<u>0.64</u>
Average	**0.44***	**0.27**	**0.26**	**0.06**	**0.32**	**<u>0.74</u>**

Another way of evaluating CPD methods is visually. Below are graphs of time series datasets, Torque3 and WOB1. The graph is shaded purple when a change point is detected, with darker shades indicating a more severe changepoint.

[4] `https://github.com/seboelter/changepoynt_IAS/tree/master`
[5] `https://pypi.org/project/changepoynt/`

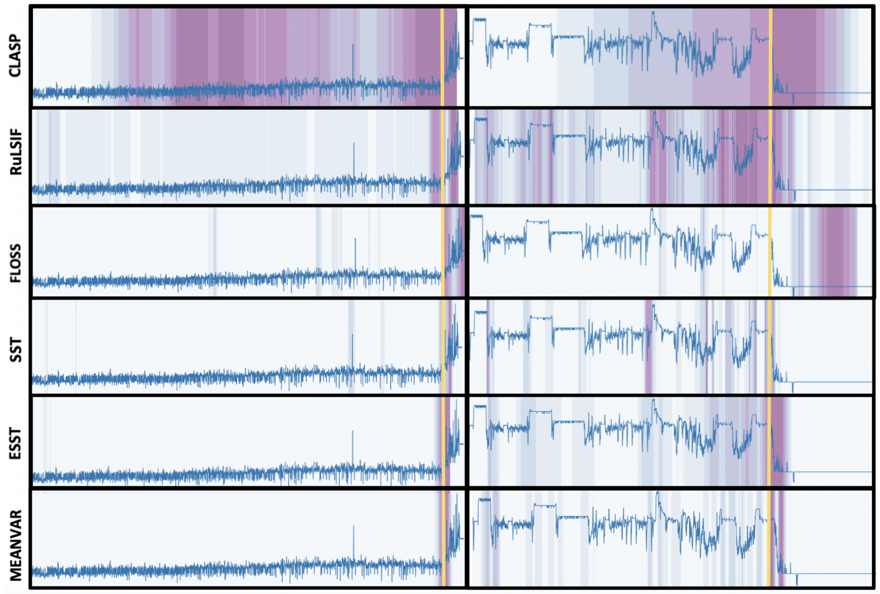

Fig. 5. A sampling of visual results from selected algorithms for two datasets. Dark background indicates the change point score of the methods. The yellow vertical line indicates a drilling fault.

6 Discussion

The results shown are promising for detecting faults. For the F1-Score, there was one fault, Torque5, where none of the methods performed well. This highlights the importance of monitoring multiple data streams simultaneously in addition to monitoring other sensors outside the scope of this work. We find that the ESST performance fits well within a comfortable threshold to be added to the greater system in place to monitor the drill's performance.

Open questions and points remain. The most obvious point is the limited amount of testing data, which is directly linked to the application scenario. With our selection of unsupervised methods, we aim to mitigate the effects of missing data, but we acknowledge that a more extensive collection of faults would enhance our analysis.

Another open question is the quantification of the detection delay, which was not discussed as a part of our evaluation. In our application, we aim to use the methods to detect faults early and retract the drill into a stable operating status, where human supervision can assess the situation. Therefore, detecting faults as early as possible is of great interest. With detection delay, we mean the difference in time or samples between the first occurrence of the fault and the latest sample required to make the detection. An extreme example would be offline methods, where the detection delay approaches infinity. For these methods, we require many more samples after the fault happens to make a detection. All the methods evaluated in this paper require the definition of some window length. Two of these windows (before and after t) are then compared to detect changes. The window length is immediately linked with the minimum detection delay; at best, the window after t only requires one faulty sample to make a detection. Realistically, we need more faulty samples to make the detection. Unfortunately, the ten signals used for evaluation in this paper are not enough to empirically measure significant results for the detection delay. So, we postpone this evaluation to future work. All things considered, our current sliding window of 90 time-steps, and our sampling rate of 14 time-steps a second, an approximately 6-second delay are reasonable for detecting faults in this scenario.

7 Conclusions

Detecting drilling faults in data recorded by specialized robotic drilling agents acting in high-risk environments requires specialized algorithms that are robust regarding noise and data sparsity. This paper shows how subspace-based change point detection methods, like the SST, fulfill the requirements for unsupervised, data and computationally efficient algorithms. We proposed a novel subspace-based method, termed the ESST, that builds on the SST and stabilizes the scoring by changing the construction of the trajectory matrix using the right singular vectors of the subspace that have been ignored before. An empirical evaluation on different real faults shows the performance in comparison to other established algorithms and how the intuitive derivation holds true in practice.

Acknowledgments. This material is based upon work supported by the Minnesota Space Grant Consortium and the National Science Foundation CSGrad4US Fellowship under Grant No. 2240197. SIEMENS Energy AG has supported this work in the context of Project CATCH. We thank Martin Buehl and Markus Zenker for their valuable insights into industrial data analytics.

References

1. TRIDENT Drill. science.nasa.gov/lunar-science/clps-deliveries/prime-1-im/
2. Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection. arXiv preprint arXiv:0710.3742 (2007)
3. Aminikhanghahi, S., Cook, D.J.: A survey of methods for time series change point detection. Knowl. Inf. Syst. **51**(2), 339–367 (2017)
4. Boelter, S., Temesgen, E., Glass, B., Gini, M.: Understanding drill data for autonomous application. In: International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications at AAMAS (2024)
5. van den Burg, G.J.J., Williams, C.K.I.: An evaluation of change point detection algorithms. arXiv:2003.06222v3 [stat.ML] (2020)
6. Cayeux, E., Daireaux, B., Ambrus, A., Mihai, R., Carlsen, L.: Autonomous decision-making while drilling. Energies **14**, 969 (2021)

7. De Ryck, T., De Vos, M., Bertrand, A.: Change point detection in time series data using autoencoders with a time-invariant representation. IEEE Transactions on Signal Processing **69**, 3513–3524 (2021)
8. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**, 211–218 (1936)
9. Ermshaus, A., Schäfer, P., Leser, U.: ClaSP: parameter-free time series segmentation. Data Mining and Knowledge Discovery **37**(3), 1262–1300 (2023)
10. Gharghabi, S., Ding, Y., Yeh, C.C.M., Kamgar, K., Ulanova, L., Keogh, E.: Matrix Profile VIII: Domain agnostic online semantic segmentation at superhuman performance levels. In: Proceedings of the IEEE ICDM. pp. 117–126 (2017)
11. Glass, B., Bergman, D., Parro, V., Kobayashi, L., Stoker, C., Quinn, R., Davila, A., Willis, P., Brinckerhoff, W., Warren-Rhodes, K., Wilhelm, M., Caceres, L., DiRuggiero, J., Zacny, K., Moreno-Paz, M., Dave, A., Seitz, S., Grubisic, A., Castillo, M., Bonaccorsi, R., the ARADS Team: The Atacama Rover Astrobiology Drilling Studies (ARADS) Project. Astrobiology **23**, 1245–1258 (2023)
12. Glass, B., Stoker, C., Battah, H., Boelter, S., Fortuin, C., King, I., Stevenson, T., Stucky, T.: Trident drill validation at Mars analog field sites. In: 55th Lunar and Planetary Science Conference (LPSC) (2024)
13. Idé, T., Inoue, K.: Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In: Proceedings of the SIAM International Conference on Data Mining. pp. 571–575 (2005)
14. Idé, T., Tsuda, K.: Change-point detection using Krylov subspace learning. In: Proceedings SIAM International Conference on Data Mining. pp. 515–520 (2007)
15. Leung, R., Hill, A.J., Melkumyan, A.: Automation and Artificial Intelligence technology in surface mining: A brief introduction to open-pit operations in the Pilbara. IEEE Robotics & Automation Magazine pp. 2–21 (2023)
16. Liu, S., Yamada, M., Collier, N., Sugiyama, M.: Change-point detection in time-series data by relative density-ratio estimation. Neural Networks **43** (2013)
17. Mohammad, Y., Nishida, T.: On comparing SSA-based change point discovery algorithms. In: IEEE/SICE Int'l Symp. on System Integration. pp. 938–945 (2011)
18. Moskvina, V., Zhigljavsky, A.: An algorithm based on singular spectrum analysis for change-point detection. Communications in Statistics-Simulation and Computation **32**(2), 319–352 (2003)
19. Schäfer, P., Ermshaus, A., Leser, U.: ClaSP – time series segmentation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. p. 1578–1587. CIKM '21, New York, NY, USA (2021)
20. Schmidl, S., Wenig, P., Papenbrock, T.: Anomaly detection in time series: A comprehensive evaluation. Proceedings of VLDB Endowment **15**(9), 1779–1797 (2022)
21. S.T.M.D., N.: Stakeholder webinar: Civil space shortfall ranking (2024)
22. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Signal Processing **167**(C) (2020)
23. Wardt, J., Cayeux, E., Mihai, R., Macpherson, J., Annaiyappa, P., Pirovolou, D.: Taxonomy describing levels of autonomous drilling systems: Incorporating complexity, uncertainty, sparse data, with human interaction. In: IADC/SPE International Drilling Conference and Exhibition (2024)
24. Weber, L., Lenz, R.: Machine learning in sensor identification for industrial systems. it - Information Technology **65**(4-5), 177–188 (2023)
25. Wu, R., Keogh, E.J.: Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. IEEE Transactions on Knowledge and Data Engineering **35**(3), 2421–2429 (2021)