Draft version May 8, 2025
Typeset using LaTeX twocolumn style in AASTeX7

Anomaly Detection in Asteroid Patterns and Trends

Savannah Chappus[1]

[1] Northern Arizona University

ABSTRACT

As part of the Solar System Notification Alert Processing System (SNAPS), which is a verified ZTF and Rubin Observatory alert broker, we are developing a computational approach to detecting anomalous behavior in individual asteroids. In this paper, we introduce the Anomaly Detection in Asteroid Patterns and Trends (ADAPT) program, designed to detect anomalous behavior in asteroids using a combination of unsupervised machine learning approaches, including Density-based Spatial Clustering of Applications with Noise (DBSCAN) and Isolation Forest. We introduce the previous literature on the topic, as well as the history of the ADAPT program, and then introduce the algorithm design. We discuss the DBSCAN and Isolation Forest results as they stand on their own, and then draw direct comparisons to the hybrid approach, which combines DBSCAN and Isolation Forest to produce more filtered results. Finally, we discuss the results of each approach, the benchmarking techniques and results, and future work for the project.

1. INTRODUCTION

As a part of the SNAPS alert broker (M. Gowanlock et al. 2024), ADAPT is gearing up to assist in Rubin Observatory's Legacy Survey of Space and Time (LSST; Ž. Ivezić et al. 2019). In anticipation of the launch of the LSST, many people have begun to develop classification and detection pipelines aimed at astrophysical sources, including, but not limited to: asteroid population outliers (M. Gowanlock et al. 2024), anomalous astrophysical transients (P. D. Aleo et al. 2024), and light-curve anomalies (K. L. Malanchev et al. 2021).

In this paper, we present a supplementary pipeline to the existing SNAPS infrastructure, which is designed to detect anomalies in individual asteroid behavior rather than detect population outliers (M. Gowanlock et al. 2024).

2. LITERATURE REVIEW

To begin, we would like to note that the vast majority of previous research on this topic has been in two different arenas; either detecting and classifying objects like ALeRCE (F. Förster et al. 2021), LAISS (P. D. Aleo et al. 2024), and SNAD (K. L. Malanchev et al. 2021), or it has focused on asteroid population asteroids as M. Gowanlock et al. (2024) does. In addition to this, most of the previously designed pipelines are still heavily involved with the images themselves, often training models on images with corresponding labels (as with ALeRCE), which is a manual process. We seek to design a pipeline that focuses on the properties apparent in the data, rather than relying on image analysis, in an attempt to increase the automation and to reduce the manual workload.

3. DATASETS

Here, we describe the dataset that is currently used in the ADAPT pipeline, which includes a measurement defined by SNAPS called *mag18omag8*.

3.1. *mag18omag8*

The `mag18omag8` database starts with the same properties for individual observations that are defined for the `SNAPShot1` data release (D. E. Trilling et al. 2023). In addition to these, the `mag18omag8` database includes the following property:

> *mag18omag8*: The number of standard deviations (σ) by which a source's *magapbig* measurement deviates from the line where *magapbig* = *magap* (*Note:* `magap` *is the magnitude of a source taken with an 8 pixel diameter aperture, and* `magapbig` *is the magnitude with an 18 pixel diameter aperture*). A 1:1 correspondence between *magapbig* and *magap* indicates that there is no additional flux detected in the larger aperture lens.

The primary reason we use this *mag18omag8* measurement is to aid in the detection of tails and comae of asteroids. If an asteroid were to present with either a tail or a coma, the *magapbig* value will exceed that of the *magap* value, resulting in a larger *mag18omag8* value.

4. TERMINOLOGY

Due to the fact that this paper addresses topics across multiple fields in astronomy and computer science, we define some key terminology used throughout the paper. In addition to the terminology defined by M. Gowanlock et al. (2024), namely *feature vector/point*, *dataset/database*, *feature space*, and *dimensionality*, we define the following additional terms:

- *Postage Stamps*: The images that are included in ZTF alert streams. These are smaller portions of the larger original image taken by the telescope, and feature the target object (in our case, an asteroid) in the center of the image.
- *Monsoon*: The cloud computing cluster that is available to researchers through our institution. More information about this cluster can be found here[2], but at the moment, we rely on the system's SLURM scheduler to determine which CPU to utilize, rather than requesting a specific one each time.
- *Time complexity*: A computer science concept to describe how long it takes to run an algorithm in worst-case conditions. More information, including plots of different complexities can be found here [3].

In addition to these definitions, we encourage the interested readers to digest the definitions for *rb* and *elong* from the ZTF Avro Schema page, as these features are readily used in this paper[4].

5. HISTORY OF ADAPT

The ADAPT project began in August 2022 during a week-long data processing bootcamp, which has expanded and evolved during the past three years. The current codebase has largely abandoned the origins of the ADAPT project, although the original code has been kept on hand for potential future development. We still include a brief outline of the original ADAPT codebase here because the questions that resulted from the initial implementation have helped guide our latest development on the project. As are the current goals of the project, the original project aimed to develop a program that would detect anomalous behavior in individual asteroids. In other words, this meant answering the following questions:

1. What defines an anomalous asteroid?
2. What feature space(s) are needed to detect these anomalies?
3. What kind of values in these feature spaces are indicative of anomalous behavior?
4. And finally, how do we design a pipeline that processes the data and detects these anomalies?

The initial code started to address some of these questions, albeit largely in a rudimentary way. We started by designing a pipeline that took data from the database and sent it through a single filter. This filter reduced the data to our desired feature space (*mag18omag8, rb, elong, H*), then proceeded to normalize the data for each feature and rank each observation. The rankings for each observation are just the normalized value with the following adjustments:

1. *rb* and *H* were ranked in reverse, where lower values got a higher score because we determined that having lower values for these features was potentially more likely to indicate anomalous behavior.
2. *elong* and *mag18omag8* were ranked with higher values getting a higher score, conversely because we determined these values were more likely to indicate anomalies when the values were higher.

An example of what this ranking system looks like in practice can be seen in Table 1.

Table 1. Example rankings for *H* and *elong* with mock normalized data.

***H* ranking**		***elong* ranking**	
Normalized Value (x)	Ranking % $(1-x)*100$	Normalized Value (x)	Ranking % $(x)*100$
0.0	100	0.0	0
0.003	99.7	0.024	2.4
...	...	...	...
0.4563	54.3	0.573	57.3
0.4778	52.2	0.591	59.1
...	...	...	...
0.9871	1.2	0.897	89.7
1.0	0	1.0	100

The current ADAPT codebase no longer includes this filtering system for multiple reasons, not limited to the following:

- The filtering system was extremely subjective. There was no real basis on which specific values or ranges of values indicated anomalies.

[2] https://in.nau.edu/arc/overview/

[3] https://en.wikipedia.org/wiki/Time_complexity

[4] https://zwickytransientfacility.github.io/ztf-avro-alert/schema.html has the definitions for the ZTF Avro Schemas

- The filtering system lacked any criteria for ensuring that the highest / lowest ranked values for each attribute were taken from the same observation. At one point, we implemented an option to filter by a minimum number of corresponding observations (i.e. the same ZTF ID for at least two attributes), but this was extremely expensive, both in time and resources.
- Post-processing for the sigma matrix was slow and inefficient, with little room for improvement.

Moving forward, we decided to forego this approach in favor of well-tested, unsupervised machine learning algorithms that we could easily implement, leaving plenty of room for fine tuning and adjusting the algorithms to suit our needs. We retain the old codebase for inspiration and potential incorporation into the pipeline in the future.

6. NEW ALGORITHM ADJUSTMENTS

After realizing the shortcomings in the original ADAPT pipeline, we began working on a new approach to detecting anomalous behaviors in individual asteroids. Through discussion with the SNAPS team, we realized that clustering algorithms showed potential for our use case. Generally, we are looking for bursts of behavior in an asteroid that deviates from the rest, hence the benefit of clustering algorithms. As part of the overhaul of the ADAPT project, we wanted to test a few machine learning algorithms and then experiment with mixing and matching them to see what worked the best. We also tried to pick algorithms that utilized different techniques in order to make the pipeline as robust as possible. After careful consideration of some common machine learning algorithms, we decided on the algorithms shown in Table 2.

Note that we include kNN in Table 2, despite the fact that it is not in the current codebase. Please see the Future Work section for our plans to incorporate kNN into the ADAPT pipeline. An additional reason for choosing these algorithms is that they all have routine implementation in Python's **scikit-learn** package, which aligns with our codebase.

During this overhaul, we also decided to drop H from our feature space because it is a fairly constant factor in an individual asteroid. Another benefit to dropping this feature, at least in the beginning, is that our feature space then has dimension 3, and thus can be displayed in a 3-dimensional plot. For the majority of the design and testing of our pipeline, we utilized a 3-dimensional feature space consisting of the parameters *elong, rb*, and *mag18omag8*.

Algorithm	**Benefits**	**In ADAPT**
DBSCAN	• Density based clustering • Robust outlier detection with arbitrarily-shaped clusters • Detects groupings of similarly-valued observations	Yes
Isolation Forest	• Handles high-dimensional data efficiently • Pairs easily with other algorithms • Randomized and fast	Yes
kNN	• Receptive to weighting features to rank importance • Simple implementation and integration	Not yet

Table 2. Explanation of benefits of relevant algorithms.

6.1. *DBSCAN*

The DBSCAN algorithm was first created by M. Ester et al. (1996) and generally, it groups points based on their distance (*eps*) to other points, and the minimum number of points (*min_samples*) needed to form a cluster, and any points that are too far away and do not meet the minimum are labeled as noise. Each point in the dataset receives a label based on the cluster that it is in, and these labels can be used later on for further filtering.

Since the ADAPT codebase is in Python, we utilized the DBSCAN implementation from Python's scikit-learn package. For our current needs, we only used the *eps* and *min_samples* parameters, which we tuned to match our data. Additionally, because our current feature space only has 3 features, we were able to plot the results, which readers can find in Appendix **??**. Currently, the pipeline fetches data from the database, trims it to the desired feature space, normalizes it (also using scikit-learn), then runs DBSCAN on the results. In our initial tests with DBSCAN, we produced additional questions to guide our development:

1. What clusters are we interested in?
2. How do we determine how many, or which clusters are interesting?
3. How do the answers to the previous two questions change when we change the parameters?

We attempted to answer the first two questions with our current feature space by examining result plots for various asteroids and the corresponding ZTF postage stamps for observations in each cluster. An example of this can be found in Figure 1 where we extracted all of the ZTF postage stamps from a given cluster in the DBSCAN results. In the figure, *eps* and *min_samples* were chosen because they yielded a good balance between the number of clusters and the percentage of noise, which was determined after testing multiple combinations of both parameters. The cluster shown in the figure was chosen as an example because it is tightly packed and distinct enough from the main cluster, characteristics that may make this cluster more appealing as a candidate for manual inspection. Unfortunately, this isn't enough to discover the clusters of interesting observations. As can be seen in the figure, there are 5 non-noise clusters for this asteroid, but we still don't know what makes one cluster more interesting than another without manually inspecting each one. This process is time-consuming and requires visual inspection of the results, which means that determining what is interesting is highly subjective.

On its own, DBSCAN provides at least one immediate benefit over the old ADAPT process: the entire process – especially the post-processing – is significantly more efficient than before. Unfortunately, DBSCAN does not reduce the amount of manual examination needed enough on its own. In preparation for the massive amount of data expected from the LSST, our goal is to move away from this manual process. Thus, we proceed to our next algorithm.

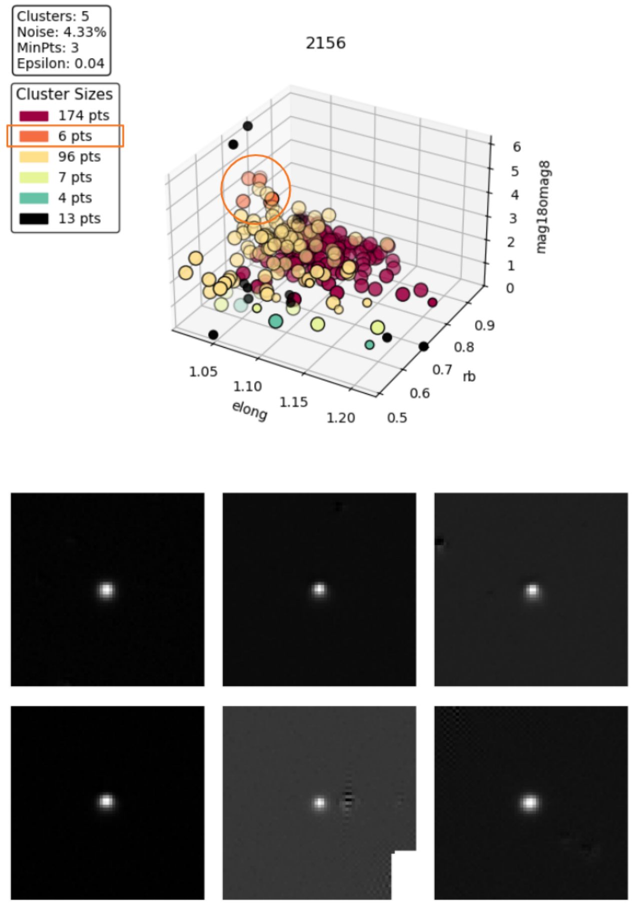


Figure 1. Example of post-processing on DBSCAN results for asteroid **2156** using *min_samples* = 3, and *eps* = 0.04.

6.2. *Isolation Forest*

The second algorithm we chose to include in our pipeline is the Isolation Forest algorithm, first implemented by F. T. Liu et al. (2008). The algorithm works by recursively partitioning the data in order to isolate anomalous points. It works under the premise that anomalous points should be distinct from the other data and thus can be separated using fewer partitions. For our codebase, we used an Isolation Forest implementation available through Python's scikit-learn package. Currently, the only parameter we use for our Isolation Forest model is the *random_state*, which is just to allow for reproducible results since Isolation Forest randomly selects features and splitting points for the data. We opt not to pre-define an outlier proportion (called *contamination* in the scikit-learn implementation) because we want to detect as much as possible now and hone our process later. Figure 2 demonstrates an Isolation Forest plot and the corresponding ZTF postage stamps for six select observations for asteroid 2156 (Kate).

On its own, Isolation Forest does not provide much benefit to our pipeline because we are faced with the same issue as with DBSCAN: how do we determine which points in the "anomaly" subset are worth looking at? While Isolation Forest does shrink the result set that would need manual inspection, it is still too large to inspect for every asteroid. Again, we need more, which leads us to our hybrid algorithm approach.

6.3. *Hybrid Approach*

There are clear benefits to both the DBSCAN and Isolation Forest algorithms, but on their own, neither really achieves what we need. Therefore, we designed our hybrid approach to combine DBSCAN and Isolation Forest to produce an even smaller set of results. In attempting to determine what the final phase of filtering should do, we asked ourselves what would make one cluster more interesting than another of similar size or distance from the main cluster. Initially, we decided that any non-noise clusters with a certain threshold of points that were also labeled by Isolation Forest as anomalies

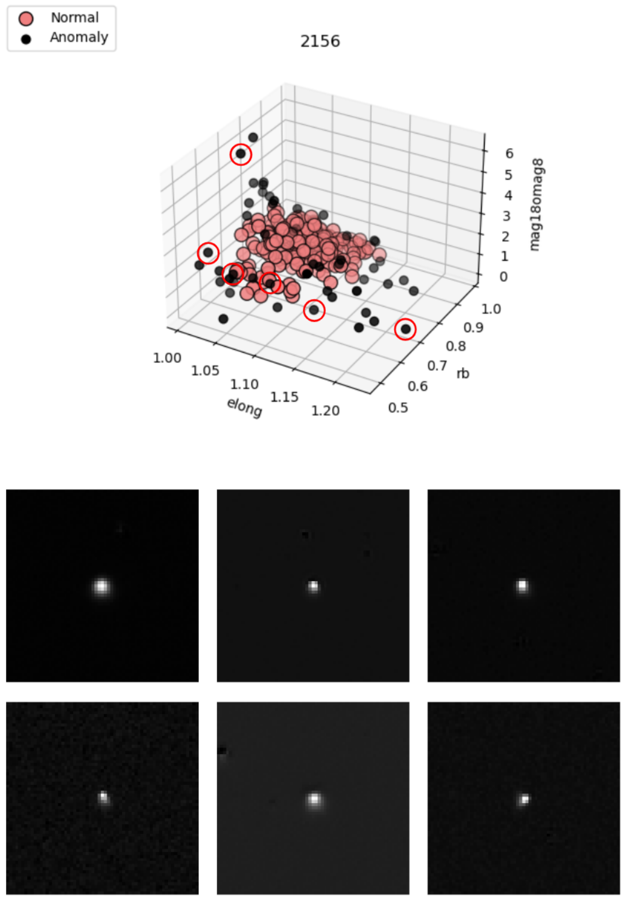


Figure 2. Isolation Forest results for asteroid **2156**.

would be more interesting than clusters with no anomalous points from Isolation Forest. Therefore, we combine the algorithms following the process described below.

> First, we take DBSCAN clusters and cross-reference them with the Isolation Forest anomaly results. We ignore clusters that are labeled as noise, are the largest in size, and any with less than 50% of their points also labeled as an anomaly by Isolation Forest. From the remaining results, we additionally ignore asteroids with less than 2 remaining clusters which fit the previous criteria.

This filtering can be seen in Figure 3, where we input four asteroids into ADAPT and only one meets all criteria and thus is output at the end. We acknowledge the following about this process:

1. These cutoffs (>50% and >2 clusters) are highly dependent on the features being used and thus require more testing and observation of the results to determine what modifications or additions need to be made.
2. Some known active asteroids (see Table 4 for examples) are not included in the final result set.
3. We have yet to test adding a kNN model to the pipeline, which will inevitably change the results and alter the final processing of the data.

Nevertheless, we proceed to our benchmarking and results given the current pipeline.

7. BENCHMARKING

In order to test the efficacy of our new pipeline, we employed several benchmarking techniques to quantify the success of our algorithms. Before introducing these techniques, we would like to note that there are few culminating lists of active asteroids to which we can compare our result set. Of the ~1.5 million, fewer than 60 asteroids have been found to be active (see Table 1 of C. O. Chandler et al. 2018), which makes verifying our results difficult. In addition to the ~60 known active asteroids, the SNAPS team has identified candidates for activity that we have included in our comparison.

7.1. *Timing*

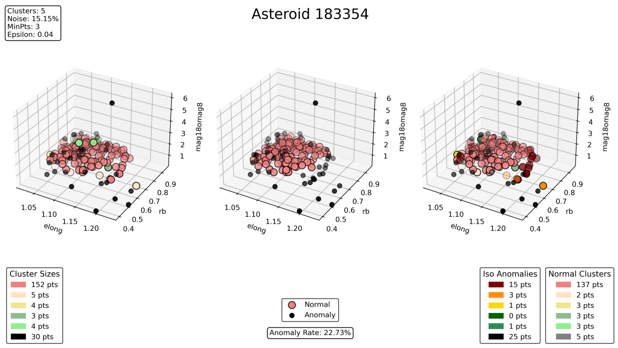
The first technique we used was simply timing each algorithm, which can be seen in Table 3. We averaged over three time trials, ensuring that we had exclusivity on the CPU on Monsoon to avoid polluting the results. We also ran each algorithm with the same 100 asteroids to ensure comparability. Note that we include the hybrid algorithm approach in this table, despite the fact that it incorporates both DBSCAN and Isolation Forest and thus is not fully comparable to each algorithm on its own; this is to demonstrate that our additional filtering adds minimal time to the overall pipeline.

Algorithm	**Avg. Time (s)**	**Time (s)/Obj**
DBSCAN	52.779	0.527
Isolation Forest	45.410	0.454
Hybrid	156.826	1.56

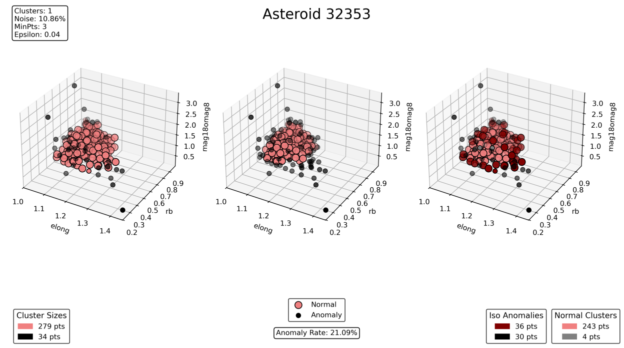
Table 3. Timing results of the three algorithms tested on 100 asteroids for the ADAPT pipeline.

7.2. *Cross-Referencing*

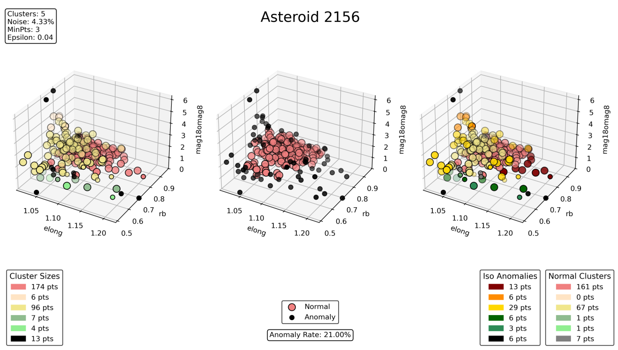
The second technique we employed to test our pipeline is how many known active asteroids it was able to detect. We compiled a list of 98 known or suspected active asteroids, taken from multiple sources, and cross-referenced our results against this list. We have included asteroids from C. O. Chandler et al. (2018) and D. Jewitt & H. H.



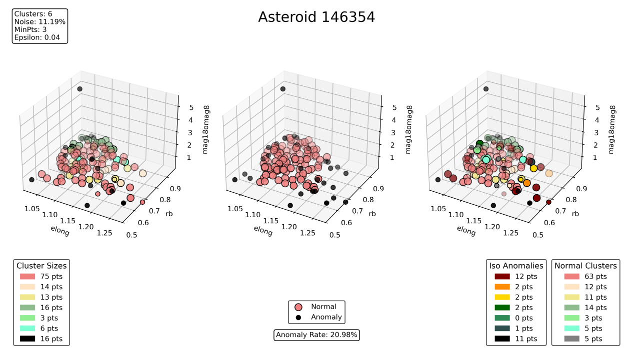
(a) **Fail:** Only has 1 cluster with >50% of their points labeled as anomalies.



(b) **Fail:** Has no non-noise and non-maximal clusters to test for anomalies.



(c) **Pass:** Has 3 clusters with >50% of their points labeled as anomalies.



(d) **Fail:** All 5 non-noise, non-maximal clusters have <50% of their points labeled as anomalies.

Figure 3. Examples of asteroids that pass/fail the final processing in the ADAPT pipeline.

Hsieh (2022), as well as an internal list of suspected active asteroids that has been compiled by the SNAPS team. The results are described in Table 4.

7.3. *Scalability*

The third and final technique that we utilized to determine the success of our pipeline is how well it will scale and how well it handles real-time data. This is where much of the future of ADAPT lies, most of which is noted in the Future Work Section. Briefly, some of the potential future bottlenecks in our pipeline are as follows.

- LSST data will be retrieved in real time, which means that clustering algorithms like DBSCAN may require additional data (probably taken from ZTF) in order to be effective.

- Currently, the ADAPT pipeline requires manually changing variables to customize each run, primarily for testing purposes, but this will need to be removed in favor of automation for LSST data.

8. RESULTS

Given the nature of the ADAPT pipeline, our results are still largely underway; however, we do have some preliminary results, which we outline in this section. First, our current database holds observations for 40,575 unique asteroids, with a rough average of 100 observations per asteroid. We ran all of these through the ADAPT pipeline without requesting exclusive resource access on Monsoon, the results of which are shown in Table 5

Images of the hybrid plots and some corresponding ZTF postage stamps for a few of the resulting asteroids can be found in Appendix A.

9. FUTURE WORK

Although much progress has been made on the ADAPT pipeline, we also want to make a special note of some potential avenues for future work on the project. We have subdivided this section into 3 general areas: LSST Prep, Further Processing, and Time and Efficiency.

Table 4. Presence of Known Active Asteroids in ADAPT Results

Object	In Results	Object	In Results
(145) Adeona	Yes	(101167) 1998 SW2	No
(493) Griseldis	No	(596) Scheila	No
(704) Interamnia	No	(779) Nina	No
(807) Ceraskia	Yes	(820) Adriana	Yes
(894) Erda	Yes	(1026) Ingrid	No
(1033) Simona	Yes	(1079) Mimosa	Yes
(1194) Aletta	No	(1236) Thais	No
(1269) Rollandia	No	(1279) Uganda	Yes
(1302) Werra	No	(1474) Beira	Yes
(1476) Cox	No	(2031) BAM	Yes
(2060) Chiron	No	(2134) Dennispalm	No
(2185) Guangdong	No	(2201) Oljato	No
(2347) Vinata	No	(2414) Vibeke	Yes
(2543) Machado	Yes	(3200) Phaethon	Yes
(3306) Byron	No	(3457) Arnenordheim	No
(3510) Veeder	No	(3552) Don Quixote	No
(3646) Aduatiques	Yes	(3958) Komendantov	No
(4015) Wilson-Harrington	No	(4376) Shigemori	Yes
(5554) Keesey	No	(6478) Gault	Yes
(7231) Porco	No	(7968) 133P/Elst-Pizarro	No
(10143) Kamogawa	No	(11790) Goode	Yes
(11987) Yonematsu	Yes	(12329) Liebermann	No
(13487) Novosyadlyj	Yes	(13698) 1998 KF35	Yes
(14204) 1990 AM20	No	(14759) 6520 P-L	Yes
(15159) 2000 FN41	No	(15165) 2000 GR89	Yes
(15484) 1999 CU46	Yes	(16945) 1998 HD46	Yes
(19001) 2000 RV60	Yes	(20857) Richardromeo	Yes
(21049) 1990 SU16	No	(24678) 1989 TR11	No
(24684) 1990 EU4	Yes	(26249) 1998 QV50	No
(26348) 1998 XO94	No	(26978) 1997 UZ4	No
(29218) 1992 AY	No	(30134) 2000 FR49	No
(30465) 2000 OY13	No	(30771) 1986 PO2	Yes
(31450) Stevepreston	Yes	(31743) 1999 JK79	Yes
(32041) 2000 JP26	Yes	(33336) 1998 VF7	No
(35101) 1991 PL16	No	(37565) 1988 VL3	Yes
(42448) 3393 T-3	Yes	(44060) 1998 FU42	No
(44351) 1998 RA79	No	(47027) 1998 VX29	Yes
(48433) 1989 US1	No	(48540) 1993 TW8	Yes
(49184) 1998 SW73	Yes	(50960) 2000 GN82	No
(53951) 2000 GC58	No	(55006) 2001 QZ24	Yes
(57341) 2001 QR263	Yes	(58192) 1992 AQ	No
(59245) 1999 CT7	Yes	(61088) 2000 LZ21	Yes
(61536) 2000 QR63	No	(62412) 2000 SY178	Yes
(65667) 1987 SM5	No	(65803) Didymos/Dimorphos	Yes
(66103) 1998 SJ24	Yes	(73768) 1994 PO10	Yes
(84528) 2002 UP9	Yes	(94986) 2001 YE118	Yes
(103501) 2000 AT245	Yes	(162173) Ryugu	No
(248370) 2005 QN173	No	(300163) 2006 VW139	Yes

Note: some of the objects in this list may not still be active, and thus our method will not be able to detect them.

Total Time (s)	50,156
Time/Asteroid (s)	1.2
Number of Asteroids Flagged	17,122 (42.19%)
Known Active Asteroids	47 (41.22%)

Table 5. Stats for the ADAPT pipeline

9.1. *LSST Prep*

Part of the point of ADAPT is to prepare for the launch of LSST, and as such, ensuring that our pipeline will handle real time data streams and be effective at processing the incoming data, we include our plans for future modifications below.

1. Modifying the pre-processing and data retrieval process to incorporate incoming data streams rather than fetching from a static database.
2. As long as ADAPT includes clustering, we acknowledge that clustering algorithms are only successful above a certain number of data points. As such, we may implement an additional pipeline to combine existing ZTF data, when possible, to the incoming LSST data so that asteroids meet the minimum number of data points needed for the clustering algorithms to be effective.

9.2. *Further Processing*

In addition to preparing for LSST data, we would like to continue refining and adding additional processing to our pipeline. Some of these ideas are listed below.

1. Adding kNN to the filtering pipeline.
2. Continuing parameter tuning for both DBSCAN and Isolation Forest.
3. Incorporating additional filtering on Isolation Forest and DBSCAN results individually before combining them and performing post-processing. This could include cutoffs for a certain noise level or number of clusters from DBSCAN or a certain anomaly ratio from Isolation Forest.

9.3. *Time and Efficiency*

Finally, we recognize that, while our processing speed is decent, there is definite room for improvement, especially in preparation for LSST data, which will need to be processed in a much faster timeline than what we currently support. Most of this speedup will come in the form of parallelizing portions of the pipeline where possible.

1. Both DBSCAN and Isolation Forest support CPU parallelization in scikit-learn with a parameter called *n_jobs*. Since Monsoon and SLURM support multi-threading on CPUS, we can easily implement this and test the speedup on ZTF data.
2. Parallelizing code for running on GPUs. In particular, DBSCAN has shown impressive speedups when parallelized for GPU processing (see M. Gowanlock et al. 2019 and M. Poudel & M. Gowanlock 2021).

10. CONCLUSION

We have begun work on a supplemental processing pipeline called ADAPT, which is designed to detect anomalous behavior in individual asteroids. We intend for ADAPT to be a downstream processing system for the Rubin-approved alert broker SNAPS, which focuses on detection and notification of anomalous solar system objects, in particular, asteroids. While the project is still well underway, we presented some preliminary results from the pipeline, including initial benchmarks to compare future improvements against.

We anticipate a significant amount of progress in the upcoming months as the SNAPS team, along with other brokers, prepares for the much-anticipated launch of LSST, and we welcome collaboration with any other alert brokers and individuals interested in the ADAPT project. For those who are interested in contributing to the ADAPT project, the codebase can be found here[5], and will be updated on a regular basis.

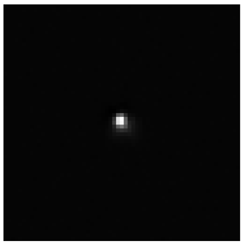
ACKNOWLEDGMENTS

We thank all the people who have made the ADAPT project what it is today. This includes, but is not limited to, Michael Gowanlock, David Trilling, Maria Chernyavskaya, and Daniel Kramer for their foundational work on the SNAPS alert broker.

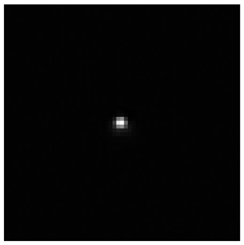
[5] https://github.com/niteflyunicorns/ADAPT

APPENDIX

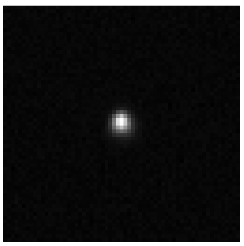
A. HYBRID PLOTS



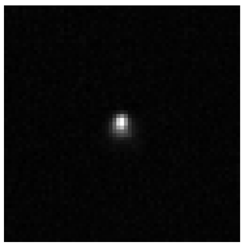
(a) ZTF22abfedcc (cluster 5)



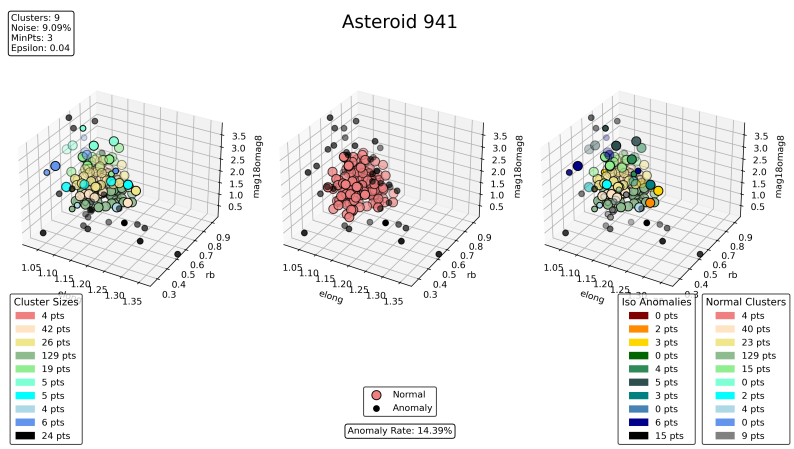
(b) ZTF22abgjoia (cluster 6)



(c) ZTF22abtdxmp (cluster 8)

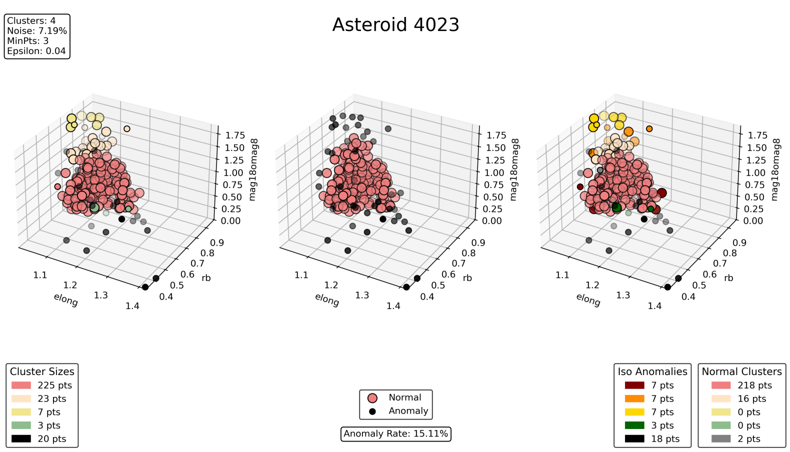


(d) ZTF22abwleee (cluster 5)

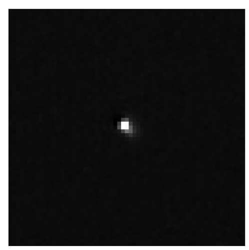


(e) Left to right: DBSCAN, Isolation Forest, and Hybrid plots

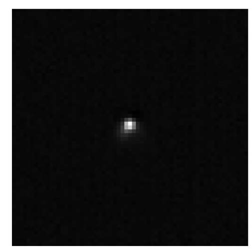
Figure 4. Hybrid results and ZTF postage stamps for asteroid **941**



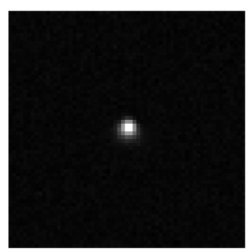
(a) Left to right: DBSCAN, Isolation Forest, and Hybrid plots



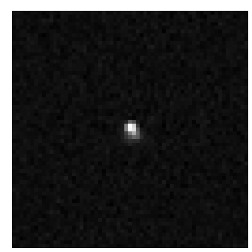
(b) ZTF21abyfmyl (cluster 2)



(c) ZTF21aceydub (cluster 2)

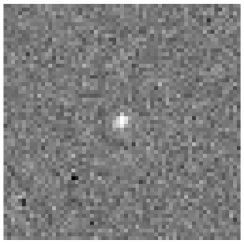


(d) ZTF21acomplv (cluster 2)



(e) ZTF23aaeyewo (cluster 3)

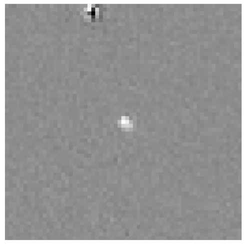
Figure 5. Hybrid results and ZTF postage stamps for asteroid **4023**



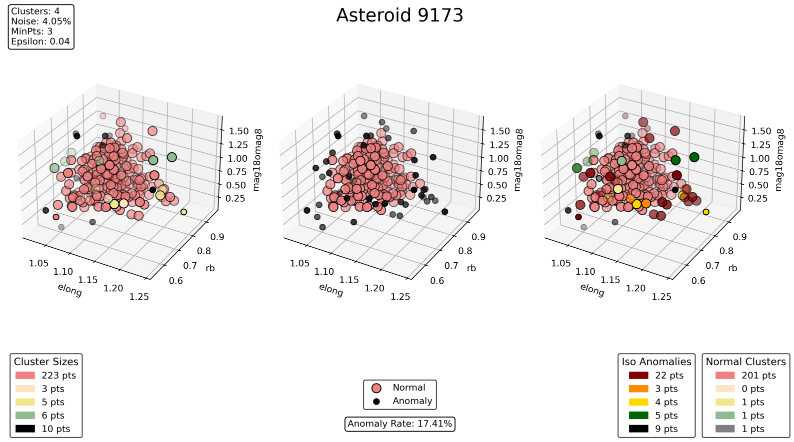
(a) ZTF18abqekih (cluster 1)

(b) ZTF22abmyffj (cluster 2)

(c) ZTF22abninzh (cluster 2)

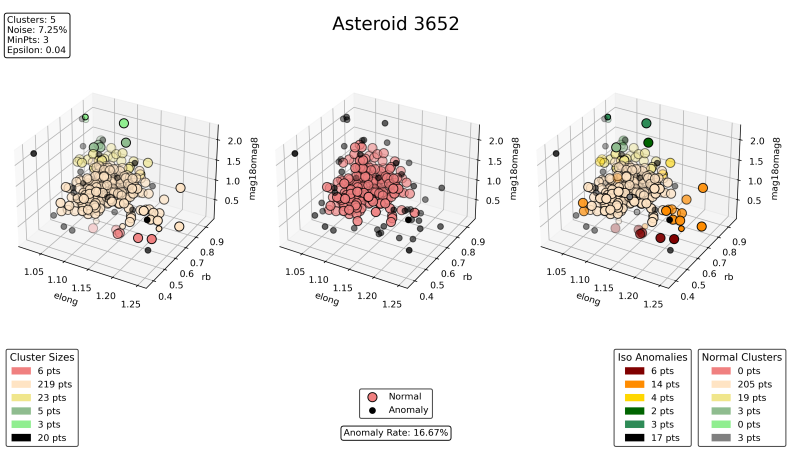


(d) ZTF23abmqvia (cluster 3)

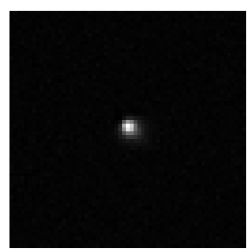


(e) Left to right: DBSCAN, Isolation Forest, and Hybrid plots

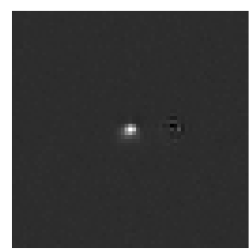
Figure 6. Hybrid results and ZTF postage stamps for asteroid **9173**



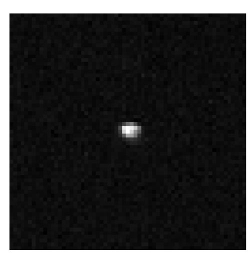
(a) Left to right: DBSCAN, Isolation Forest, and Hybrid plots



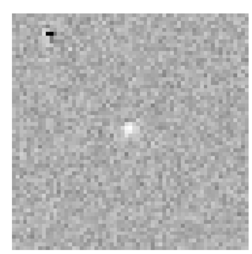
(b) ZTF21abxafwz (cluster 4)



(c) ZTF21acapkgt (cluster 4)



(d) ZTF21acibdgq (cluster 0)



(e) ZTF22abehulu (cluster 0)

Figure 7. Hybrid results and ZTF postage stamps for asteroid **3652**

REFERENCES

Aleo, P. D., Engel, A. W., Narayan, G., et al. 2024, arXiv, doi: 10.48550/ARXIV.2404.01235

Chandler, C. O., Curtis, A. M., Mommert, M., Sheppard, S. S., & Trujillo, C. A. 2018, Publications of the Astronomical Society of the Pacific, 130, 114502, doi: 10.1088/1538-3873/aad03d

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. 1996, in Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD) (AAAI Press), 226–231. https://cdn.aaai.org/KDD/1996/KDD96-037.pdf

Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., et al. 2021, The Astronomical Journal, 161, 242, doi: 10.3847/1538-3881/abe9bc

Gowanlock, M., Rude, C. M., Blair, D. M., Li, J. D., & Pankratius, V. 2019, IEEE Transactions on Parallel and Distributed Systems, 30, 766–777, doi: 10.1109/tpds.2018.2869777

Gowanlock, M., Trilling, D. E., Kramer, D., Chernyavskaya, M., & McNeill, A. 2024, The Astronomical Journal, 168, 56, doi: 10.3847/1538-3881/ad4da5

Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, The Astrophysical Journal, 873, 111, doi: 10.3847/1538-4357/ab042c

Jewitt, D., & Hsieh, H. H. 2022, arXiv, doi: 10.48550/ARXIV.2203.01397

Liu, F. T., Ting, K. M., & Zhou, Z.-H. 2008, in 2008 Eighth IEEE International Conference on Data Mining (IEEE), 413–422, doi: 10.1109/icdm.2008.17

Malanchev, K. L., Pruzhinskaya, M. V., Korolev, V. S., et al. 2021, Monthly Notices of the Royal Astronomical Society, 502, 5147–5175, doi: 10.1093/mnras/stab316

Poudel, M., & Gowanlock, M. 2021, in 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC) (IEEE), doi: 10.1109/hipc53243.2021.00049

Trilling, D. E., Gowanlock, M., Kramer, D., et al. 2023, The Astronomical Journal, 165, 111, doi: 10.3847/1538-3881/acac7f