A Survey of Scientific File Format Distributions Across Data Repositories

Sreeharshini Ambatipudi
Dulles High School
Houston, TX, USA
Email: harshiniambatipudi@gmail.com

Suren Byna
The Ohio State University
Columbus, OH, USA
Email: byna.1@osu.edu

***Abstract*— Scientific file formats are included in datasets hosted in public data repositories. They are used to store data for long-term preservation and to facilitate data sharing, but their prevalence across repositories remains largely undocumented. When analyzing such repositories, a relevant question is: "What is the distribution of file formats across repositories?" In this paper, we will examine file format distributions in four domain-specific repositories–Data.gov, ESS-DIVE, IEEE DataPort, and Hugging Face–and showcase patterns and differences across research domains. We introduce the *Scientific Data File Format Analyzer*, an open-source tool for retrieving file format distributions from scientific data repositories using both web scraping and API-based approaches. The tool visualizes normalized file formats as bar graphs, displaying the nine most prevalent formats explicitly while grouping others into an"other" category. In addition, when file size information is available in repository metadata (as in ESS-DIVE and Hugging Face), the tool retrieves this data and presents it as boxplots to characterize size distributions. Our analysis reveals distinct format preferences that reflect disciplinary workflows: XML dominates Data.gov (70% of files) due to government interoperability standards, CSV leads in ESS-DIVE (25%) and IEEE DataPort for tabular scientific data, while Parquet files are most prevalent in Hugging Face (25%) reflecting modern machine learning practices.**

GENERAL TERMS / KEYWORDS

Scientific File Formats, Data.gov, ESS-DIVE, IEEE DataPort, Hugging Face

I. INTRODUCTION

Scientific file formats are digital data formats that are designed to store and exchange scientific information. With research becoming increasingly collaborative online, scientific file formats play a crucial role in facilitating data sharing between researchers across different institutions and disciplines. These formats enable efficient storage and organization of research data, often incorporating metadata that describes the structure and content to increase interoperability [1].To support this collaborative research environment, specialized data repositories have emerged to aggregate and distribute these datasets, making scientific data discoverable and accessible to the broader research community. For example, Hugging Face, a data repository that aggregates datasets related to ML/AI applications, hosts more than one million models, datasets, and apps on its platform with tens of thousands of users globally, demonstrating the massive scale and importance of modern data-sharing infrastructure in the field of AI [2]. In this study, we present a survey of file format distributions across four major data repositories: Data.gov for government data, ESS-DIVE for environmental science, IEEE DataPort for engineering, and Hugging Face for machine learning.

These repositories were chosen as they cover a broad spectrum of research areas, which allows the tool to be tested across diverse scientific communities. Additionally, they differ in how their data can be accessed: ESS-DIVE and Hugging Face provide structured APIs, while Data.gov and IEEE DataPort mainly share metadata through web catalog pages that require scraping and parsing.

Even though these repositories are widely used, there hasn't been much written about how file formats are actually used across them. Gaining a clearer picture of that distribution could help reveal which formats researchers in different domains tend to prefer. This, in turn, can reveal what particular communities prioritize in terms of data sharing and reuse. For example, analyzing the most common file formats on Hugging Face, a repository widely used in AI/ML, provides insight into the preferences and practices of users within that field. Existing studies often focus on tracking file format distributions within a singular data repository. For example, Rimkus et al. (2014) analyzed file format policies across institutions affiliated with the Association of Research Libraries, examining the levels of confidence placed in different formats for image, text, audio, video, and other data types within institutional repositories [3]. Similarly, Peters et al. (2017) conducted a case study specifically focused on Zenodo, analyzing the characteristics and reception of different types of data records uploaded to that single repository [4]. These types of studies provide a deeper understanding of a data repository by analyzing its characteristics and the users it supports, but do not guide larger-scale trends in relation to other data repositories. Understanding the current landscape of file format usage provides a valuable baseline for the research community, as it may inform repository developers and help researchers understand the technical ecosystem of their field relative to others. The Scientific Data File Format analyzer is implemented as a modular framework focused on addressing key limitations of existing approaches that rely on repeated web scraping and makeshift analyses based on a small sample of data. The tool uses checkpoints to store retrieved metadata in JSON files, reducing repeated queries and improving the efficiency of the retrieval process. Its

design is based on an abstract class named DataRepository. This was preferred as it allows for the integration of various repositories through the generalization of certain methods rather than the creation of individual scraping files for each repository added. Another key reason for this tool is that it can support both API calls and web scraping to extract metadata (file formats, counts, and file sizes). Additionally, the tool allows for file format normalization, consisting of aggregating synonymous extensions (e.g., json and jsonl) and filtering out non-scientific file formats (.zip, .gz)

II. Background

A. Data.gov:

Data.gov is the United States government's central open data platform, launched in 2009, and is managed by the U.S. General Services Administration (GSA). It provides unified access to datasets published by federal agencies across domains such as energy, environment, health, and transportation. The catalog is populated through automated metadata harvesting from participating agencies, as mandated by the OPEN Government Data Act. As of 2025, this repository consists of contributions from over 100 federal agencies and more than 200 publishing organizations (e.g, DOE, EPA, NOAA, NASA, and USGS) along with selected state and local governments. These contributors add their catalogs through a CKAN-based infrastructure that supports large-scale metadata integration. [5] Data.gov currently hosts over 300,000 datasets and aims to make government more open and accountable by providing public access to government data for research and informed decision-making. [6] Although the repository does display the overall counts of file formats, we must filter these manually, and no breakdown of scientific formats is provided.

B. ESS-DIVE

The Environmental System Science Data Infrastructure for a Virtual Ecosystem (ESS-DIVE) is a data repository for research related to Earth and environmental sciences [7]. Hosted at Lawrence Berkley National Laboratory, the repository currently hosts more than 1,200 public datasets and has over 400 contributors from Department of Energy national laboratories, universities, and field research programs. [7]. Its datasets span various domains, including hydrology, soil biogeochemistry, ecosystem modeling, and terrestrial ecology. ESS-DIVE serves as a long-term archive for the AmeriFlux Network, a coordinated system comprising over 110 active sites across the continent that measure the exchanges of carbon, water, and energy between ecosystems and the atmosphere. [8] Similar to Data.gov, ESS Dive does not provide statistics on specific scientific file formats or their prevalence, leaving a gap in understanding the distribution of such formats across the repository.

C. IEEE DataPort

IEEE DataPort is a data repository that hosts scientific data from a wide variety of disciplines. The platform is developed by the Institute of Electrical and Electronics Engineers (IEEE) and supports scientific datasets across a range of domains, including computer science, environmental science, and biomedical research. Since its launch in 2016, it has grown into one of the largest data-sharing platforms, hosting more than 9,000 datasets [9]. While IEEE DataPort documents a set of supported file formats (e.g.,`csv`, `json`, `xml`), it does not provide counts associated with these formats. Moreover, the repository does not offer insight into the distribution of scientific file formats.

D. Hugging Face

Hugging Face is an open platform and community-driven data repository for the machine learning and artificial intelligence community. The platform hosts datasets, models, and applications for machine learning research and development. Through its hub, users can host and access datasets, pre-trained models, and evaluation benchmarks across domains, including natural-language processing, computer vision, and audio processing. [10]. As of 2025, the repository hosts over 400,000 datasets and 600,000 machine learning models. The repository offers limited statistics on file formats. It includes counts for formats such as Parquet, Arrow, CSV, JSON, and text, while other formats are grouped into broader categories (e.g.,`imagefolder`, `soundfolder`, `webdataset`), which hides the specific file types and their distributions within those folders. As a result, only partial insight into file format usage is available. Thus, it is up to the user to filter out the formats to gather the scientific file formats.

III. Methodology

The `Scientific File Format Analyzer` is implemented around an abstract `DataRepository` class, which defines an abstract method, `get_repository_metadata`. This design ensures that repository-specific logic is limited to lightweight subclasses, while shared functionality consisting of checkpointing, normalization, visualization, and data validation remains consistent across all repositories. The framework employs a hybrid retrieval approach. For repositories with structured APIs (ESS-DIVE, Hugging Face), metadata is collected via programmatic calls. For repositories lacking APIs (IEEE DataPort, Data.gov), metadata is extracted through structured web scraping using `SyncPlaywright`. We selected Playwright over Selenium due to its superior performance characteristics: faster execution times, more reliable handling of dynamic content, better resource management, and improved stability when processing large numbers of web pages sequentially.

A. Initialization

The tool automatically creates a set of folders to organize and store files used to collect metadata from each data repository. This structure helps ensure that the analysis is reproducible and makes it easier to restart or verify results if needed. The tool produces three main JSON files and one text file:

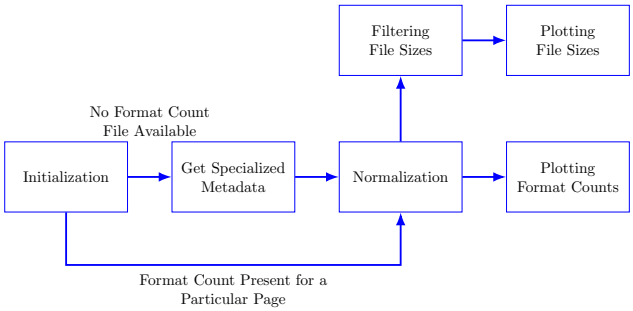

Fig. 1. File format fetching workflow for a sample data repository.

1) **Format Counts File:** Saved in the `format_counts/` directory (repository_file_format_counts.json), this JSON file contains the final aggregated results for each repository file format counts and corresponding file size data.
2) **Checkpoint File:** Stored in the `checkpoints/` directory (repository_file_format_count_page_{page}.json), this JSON file records cumulative progress for each repository. It allows the tool to resume retrieval from the most recent page-level checkpoint in case of interruptions.
3) **Repository Metadata File:** Saved in the `repository_metadata/` directory (repository_metadata.json), this JSON file contains dataset-level metadata, which includes package or dataset names, file formats, sizes, and per-dataset execution time. These entries allow for validation of format counts and file sizes against repositories. These values can be further visualized for frequency distribution plots to evaluate the performance of the tool across different repositories.
4) **Data Validation:** Saved in the `data_validation/` directory (`repository_validation_results.txt`), this text file summarizes repository-level comparisons and verification results, providing transparency across the entire analysis process.

B. Getting Repository Metadata

After initialization, the tool retrieves file format counts and file sizes. To accomplish this, each repository must implement the abstract `get_repository_metadata` method. This is invoked by the tool and must be defined by each repository-specific subclass, whether that means retrieving data through an API or by web scraping. The specific implementations for individual repositories are described below:

Catalog Dataset: The tool retrieves file format counts from Data.gov with the Playwright library. The method navigates to the repository `https://catalog.data.gov/dataset/` and extracts file format information under the tag `formats` available in the left sidebar. Both the file format labels and their associated counts are parsed and stored in a JSON file specific to each repository under `format_counts/` directory repository for further analysis and visualization. Since all counts can be retrieved in a single step, the use of page-level checkpoints is not required.

ESS-DIVE: For ESS-DIVE , the tool retrieves file format counts and sizes using ESS-DIVE REST API, which provides structured JSON responses containing dataset metadata. It first queries the `GET /packages` endpoint with pagination parameters (`isPublic=True`, `pageSize`, `rowStart`) to obtain successive batches of dataset package identifiers. For each package identifier, the tool then queries the `GET /packages/{id}` endpoint to retrieve file formats and corresponding file sizes from the repository's metadata. Retrieval continues until the API returns a batch smaller than the requested page size, indicating that all available datasets have been collected. During collection, the tool employs the use of checkpoints to ensure that data collection can occur from the most recent page in the event of interruptions and stores data in its corresponding checkpoints folder. After all data has been retrieved, the tool saves the final aggregated file format counts and file sizes in the `format_counts/` directory under a JSON file specific to each repository.

Hugging Face: For Hugging Face, the tool retrieves file formats and file sizes using ugging Face Hub API (`HfApi`), a Python client library that provides programmatic access to metadata via the Hugging Face REST API. The tool first calls the `list_datasets` method to enumerate available datasets and then uses the `dataset_info` method to obtain detailed metadata for each dataset. Within the returned metadata, the tool iterates over the `siblings` field, where each file entry includes a relative filename (`rfilename`) and, when available, a `size` attribute representing the file's content size. The tool reads this endpoint to extract file formats and corresponding file sizes from the metadata. Pagination is handled using the `limit` and `offset` parameters, enabling systematic retrieval of large collections. Similar to ESS-DIVE, the method employs page-level checkpoints, storing intermediate results in the `checkpoints/` directory, while the final aggregated format counts and file sizes are written to the `format_counts/` directory.

IEEE DataPort: For IEEE dataport, file format counts are retrieved using structured web scraping with the Playwright library. Playwright was chosen over Selenium due to its superior performance, including faster execution times, more reliable handling of dynamic web content, improved resource management, and greater stability when processing large numbers of sequential pages. Dataset listings are accessed iteratively through catalog pages using URLs of the form https://ieee-dataport.org/datasets?page=\{current_page\}. For each dataset, file format information is extracted from the `Data Formats` section of the listing page. The tool employs page-level checkpointing. Once all pages are processed, the final aggregated format counts are stored in the repository's summary file located in the `format_counts/` directory. File sizes are not available for this repository, as the corresponding metadata are not exposed through the dataset listings.

In addition to the checkpoint JSON files and the final

aggregated summary files, the tool stores raw per-dataset metadata in a JSON file. These records include the dataset or package name, file names with extensions, reported file sizes, and execution time per dataset. They are used to check data accuracy and evaluate execution performance. A typical entry in the metadata file looks like this:

```
{
  "ess-dive-3e2f8b427ee0700-165137369": {
    "formats": {
      "csv": {
        "count": 2,
        "sizes": [
          1.4462890625,
          1.849609375
        ]
      }
    },
    "execution_time_sec": 1.95
  }
}
```

C. Normalization

File extensions are normalized by converting all suffixes to lowercase and applying a predefined alias mapping to standardize equivalent labels (generic `excel` labels are mapped to `xlsx`, `text` is mapped to `txt`). Synonymous scientific formats are aggregated to ensure consistent representation across repositories. For example, `h5`, `hdf5`, `h5ad` are mapped to `hdf5`. Similarly, `json` and `jsonl` are unified under the single label `json` to maintain consistency across repositories. The tool maintains an internal collection of excluded non-scientific file formats. Since the primary goal of the framework is to analyze scientific file format distributions, these non-scientific types are filtered during normalization using this collection. For instance, compression-oriented extensions such as `.zip`, `.gz`, and `.z`, as well as descriptive or documentation-based formats like `.yml` and `.doc`, are excluded because they do not directly contain structured or experimental scientific data.

D. Filtering File Sizes

In most repositories, file sizes are provided together with file format information through the same API call in `get_repository_metadata` method. The tool applies the same normalization procedure to file sizes as it does to file format counts. After filtering out non-scientific and compressed formats, file size information is grouped by normalized format names. Specifically, all file sizes associated with a given format are concatenated into a single list using the command `filtered_sizes.setdefault(format_name, []).extend(sizes)` This approach ensures that every scientific file format, such as `csv` or `json`, accumulates the size values of all files across datasets within that repository. The result is an aggregated collection of size lists per format, which can later be used to generate box plots. For repositories that include file size metadata, such as ESS-DIVE and Hugging Face, these values are extracted directly from metadata fields (measured in kilobytes).

E. Plot File Formats

Normalized file format counts are visualized through bar charts created with Matplotlib. For each repository, the nine most frequently used scientific file formats are shown individually, and all remaining formats are grouped into an "other" category. Limiting the visualization to the most common formats highlights major trends in file usage while maintaining a clear and readable layout. Including the "other" category ensures that the full distribution is still represented, allowing straightforward comparisons of format prevalence across repositories without the clutter caused by numerous low-frequency formats.

F. Plot File Sizes

Box plots of file sizes were produced using Matplotlib to visualize the distribution of file sizes among the top nine scientific file formats in each repository in kilobytes. The plots display the range and central tendency of file sizes by indicating minimum, lower quartile, median, upper quartile, and maximum values. Focusing on the nine most frequent formats provides a clear view of size variations among the most commonly used scientific formats while keeping the figures readable.

G. Limitations and Considerations

Our analysis is limited to publicly accessible datasets available at the time of data collection, and results may change as repositories evolve and incorporate new content. Metadata availability varied across repositories, which restricted certain aspects of our analysis. For example, individual file size information was not provided by Data.gov and IEEE DataPort, limiting direct cross-repository comparisons of size distributions. To ensure feasible data collection timelines, the tool restricted retrieval to subsets of available data: the first 960 of over 9,000 datasets on IEEE DataPort, and the first 30 of 100 pages on Hugging Face. In the case of Hugging Face, API tokens were required to access metadata, and retrieval was subject to rate limits imposed by the service (1,000 API requests per five-minute period). Higher request volumes would require a paid account tier, as documented in the Hugging Face Hub rate limit policy.

H. Extension of the Tool onto a New Repository

The tool is designed to be easily adaptable to additional data repositories. For a new repository, one only needs to create a subclass of the abstract `DataRepository` class and implement the core method `get_repository_metadata`. Depending on the repository's structure, this method can retrieve metadata either through API calls (when a structured programmatic interface is available) or through web scraping (when file format and size information is only available on catalog pages). Once implemented, the new repository integrates into the existing framework and has access to all shared functionalities such as page-level checkpointing, final

aggregated checkpointing, normalization, and visualization. This modular design allows new repositories to be added without altering the core codebase.

IV. Evaluation

All experiments were conducted on a local workstation running a 64-bit operating system (Windows 10), with an x64-based processor and 16.0 GB RAM (15.7 GB usable). Development and execution were performed using the PyCharm 2023.3.4 Community Edition environment, which provided an integrated platform for coding, debugging, and visualization.

A. File Format Distributions

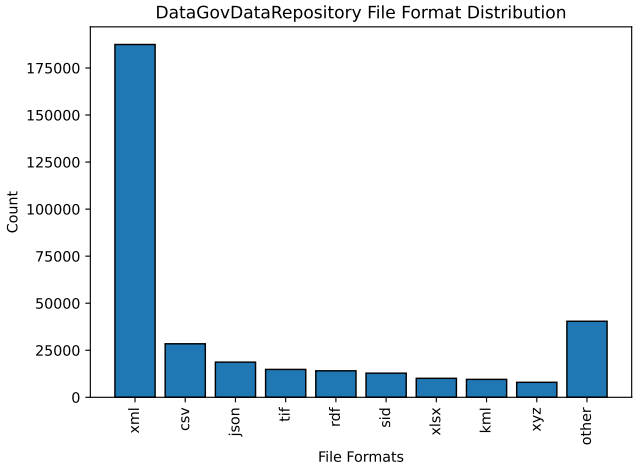

Fig. 2. Top 9 Scientific File Formats Across All Datasets within Data.gov - 355,953, Sorted by Most Popular.

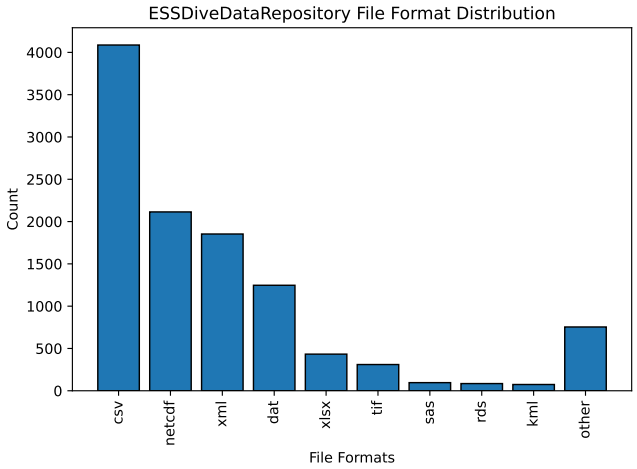

Fig. 3. Top 9 Scientific File Formats Across All Datasets within ESS-Dive – 1,322, Sorted by Most Recent.

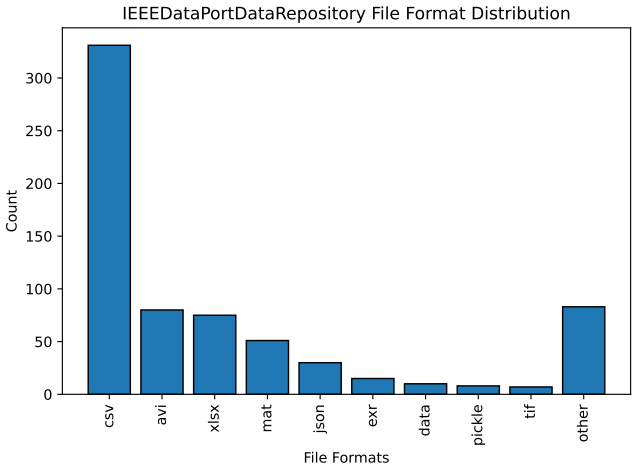

Fig. 4. Top 9 Scientific File Formats Across the First 40 out of 411 Pages in IEEE DataPort, Sorted by Most Recent.

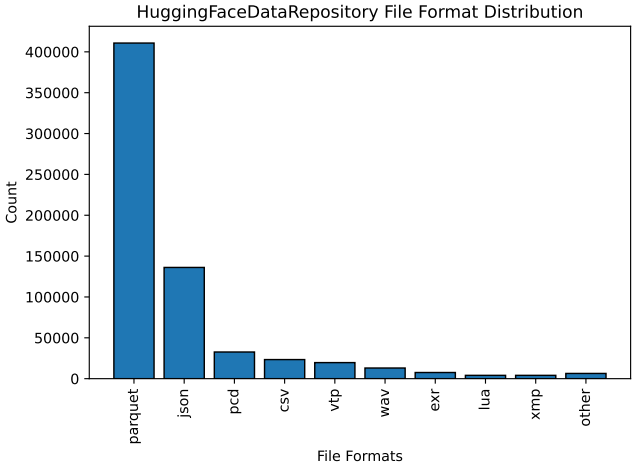

Fig. 5. Top 9 Scientific File Formats Across the First 30 Pages out of 100 in Hugging Face, Sorted by Trending.

B. File Size Distributions

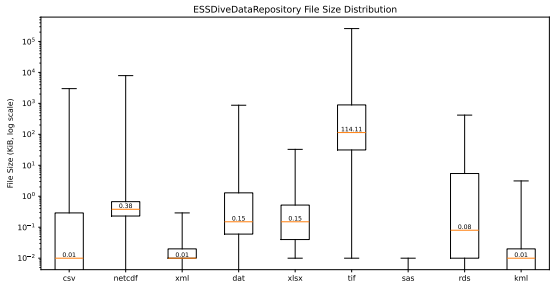

Fig. 6. The File Size Distribution in Kilobytes for the Top 9 Scientific File Formats in ESS-DIVE Represented as a Boxplot with the Median Being Highlighted

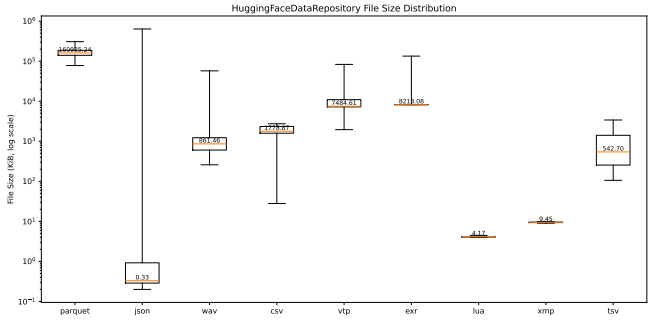

Fig. 7. The File Size Distribution in Kilobytes for the Top 9 Scientific File Formats in Hugging Face Represented as a Boxplot with the Median Being Highlighted

C. Observations

File Format Distributions: Figure 2 shows the file format distribution for the Data.gov data repository. `xml` is overwhelmingly the most common file format, representing approximately 7/10 of all files, followed by `csv` and `json` at much lower frequencies. This dominance of `xml` reflects government data standards and interoperability requirements, where XML is preferred for structured metadata exchange and regulatory compliance across federal agencies.

Figure 3 shows the file format distribution for the ESS-DIVE data repository. `csv` files represent the largest share at approximately 1/4 of all files, followed by `nc` (NetCDF) at about 1/4 of the total and by `xml` at roughly 1/8. This distribution reflects environmental science workflows where CSV is preferred for tabular sensor data and field measurements, while NetCDF serves as the standard for multidimensional climate and atmospheric data that requires self-describing metadata.

Figure 4 shows the file format distribution for the IEEE DataPort repository. `.csv` files represent the largest share at approximately 1/3 of all files, followed by `.avi` and `.xlsx`.

Figure 5 shows the file format distribution for the Hugging Face repository `.parquet` files dominate with approximately 1/4 of all files, followed by `.json` image files at about 1/10 of the total. Within this file format, .jsonl makes up half of all the JSON files. The next most common format is `.pcd`, which is used for storing 3D point cloud data in perception and computer vision applications. This distribution reflects modern machine learning workflows where Parquet provides efficient columnar storage for large datasets, compressed archives facilitate data set distribution, and PCD supports 3D perception and spatial understanding tasks central to autonomous vehicles and other modern AI applications.

File Size Distributions: Figure 6 shows the file size distribution across files in the ESS-DIVE repository by format. SAS files are among the smallest with median sizes below .01 KB, while CSV and TIFF formats exhibit variation ranging from kilobytes to gigabytes, with TIFF files reaching over 100 GB for high-resolution geospatial imagery and CSV files extending to over 1 GB for large tabular datasets. This pattern reflects the diverse nature of environmental data where XML serves primarily for lightweight metadata and configuration files, while CSV and TIFF formats accommodate everything from small sensor measurements to massive climate datasets and satellite imagery.

Figure 7 shows the distribution of file sizes across formats in the Hugging Face repository. `parquet` files have the largest median size, at around 100 megabytes, reflecting their role in storing large structured datasets. In contrast, `json` files are the smallest, with median sizes around 0.33 KiB, consistent with their use for lightweight metadata and configuration storage.

V. Validation of Tool

Accuracy of the tool's outputs was assessed using the JSON files stored in the `repository_metadata/` directory. Each file records dataset or package identifiers as keys and maps them to their associated file formats, counts, and reported file sizes when available. To evaluate each repository, we did 3 sets of validations shown below:

Validation 1: Using the raw per-dataset metadata JSON files, we selected a sample of file formats and sizes to manually verify them against the corresponding repository entries. For example, in the ESS-DIVE repository, the metadata for a specific package ID records the file format, the number of files, their sizes, and the total execution time. An example entry is shown below:

```
"ess-dive-b541bd3a784735c-
20251002T205839707": {
    "formats": {
      "xml": {
        "count": 1,
        "sizes": [14.85]
      }
    },
    "execution_time_sec": 3.08
  }
```

This example illustrates how the tool captures key information about each dataset, allowing for both accuracy checks and performance evaluation.

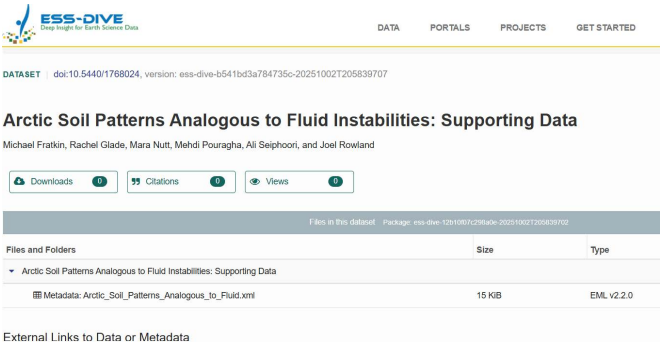

Fig. 8. Screenshot showing an example of the raw per-dataset metadata JSON file used for verification.

Validation 2: The tool verifies consistency between the reported file-format counts and the number of file-size entries

in the raw per-dataset metadata JSON files. For example, in the ESS-DIVE repository, the metadata for a specific package ID contained:


```
"ess-dive-0c4579ec3e4bbad-20250618T163845979": {
  "formats": {
    "zip": {
      "count": 2,
      "sizes": [
        7.0068359375,
        117167.1689453125
      ]
    },
    "xml": {
      "count": 1,
      "sizes": [
        58.8056640625
      ]
    }
  },
  "execution_time_sec": 1.91
```


Fig. 9. Screenshot showing an example of the raw per-dataset metadata JSON file used for Validation 2.

Here, the tool confirmed that there were nine CSV files and nine corresponding size entries, indicating that the data were recorded correctly. All matches and mismatches detected during this procedure were written to a validation log (plain-text).

Validation 3: We compared the top nine scientific file-format counts with the raw per-dataset JSON data after aggregation and filtering. This check ensured that the format distributions shown in the visualizations were consistent with the underlying dataset-level records. Any discrepancies were logged in the same validation file, with each check marked as "MATCH" or "MISMATCH."

A. Time - Frequency Distribution

In addition to metadata, the tool also records the execution time required to fetch and process each dataset. Specifically, `field` is appended to every dataset entry in the JSON file, capturing the elapsed time in seconds for that dataset's retrieval and parsing. This helps in viewing performance at the dataset level and allows us to analyze variability in access times across repositories.

B. Analysis of Time Frequency Distribution

To further evaluate the efficiency of our tool, these execution times were aggregated across datasets and visualized as frequency distribution plots. Catalog Dataset does not have this distribution, as all file formats are retrieved simultaneously. Among the repositories analyzed, IEEE Dataport exhibits the greatest variance in execution time, with retrieval durations spanning a wide range and the highest median time overall. In contrast, Hugging Face has the shortest median retrieval time, with most datasets being accessed almost instantaneously and a very limited spread in the execution time

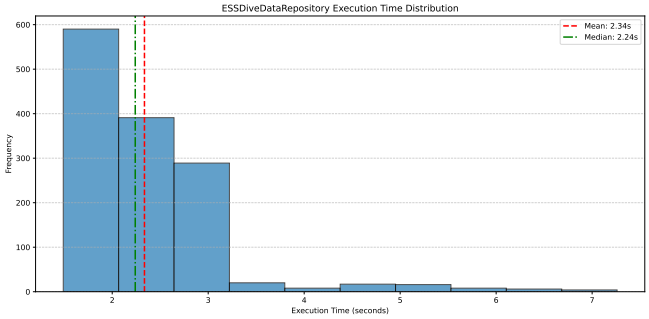

Fig. 10. ESS Dive Time Frequency Plot Across all Datasets Sorted by Most Recent

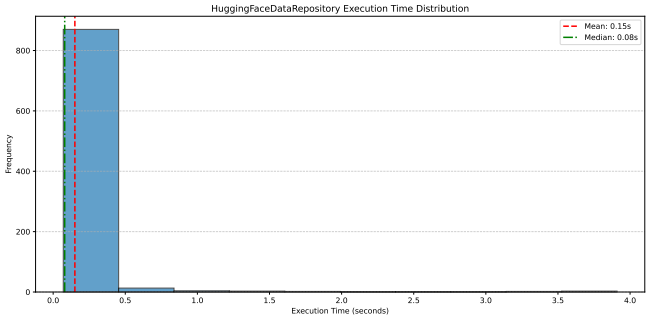

Fig. 11. Hugging Face Time Frequency Plot Across the First 30 Pages Sorted by Trending.

distribution. ESS-DIVE falls between these two extremes, showing a moderately right-skewed distribution, where the majority of retrievals complete quickly but a few outliers require longer access times.

VI. Conclusion

This study presents a comparative analysis of file format distributions across four major domain-specific data repositories, revealing distinct patterns that reflect the unique needs and workflows of their respective research communities.

The dominance of XML in government data (Data.gov) reflects regulatory compliance and interoperability standards essential for federal data exchange. Environmental sciences (ESS-DIVE) show a preference for CSV and NetCDF formats that accommodate both simple tabular sensor data and complex multidimensional climate datasets. The diverse format landscape in IEEE DataPort illustrates the broad scope of modern scientific research, showing wide preferences across multimedia formats. Meanwhile, Hugging Face's emphasis on Parquet files signals the machine learning community's adoption of modern columnar storage optimized for large-scale data processing.

While this analysis focused on frequency distributions, future work could explore data volume patterns, temporal format adoption trends, and relationships between format choices and data reuse. As scientific data sharing grows in importance, understanding these format ecosystems becomes critical for building sustainable, interoperable research infrastructure.

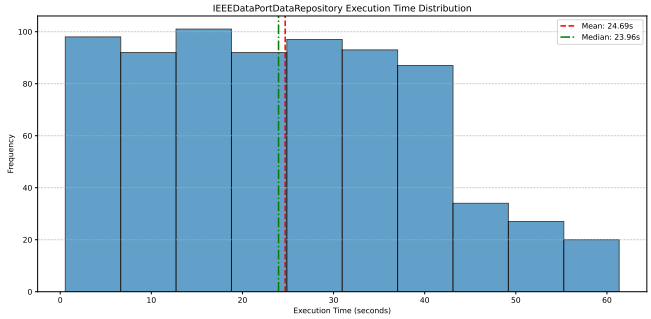

Fig. 12. IEEE DataPort Time Frequency Plot Across the First 40 Pages Sorted by Most Recent.

The code and data used in this study are available at: `https://github.com/idtlab/`. We encourage others to reproduce our findings and welcome contributions that analyze file format distributions in other domain-specific repositories.

REFERENCES

[1] Robert E McGrath. "XML and Scientific File Formats". In: *2003 Seattle Annual Meeting*. Vol. 337. Citeseer. 2003.

[2] Leigh Mc Gowran. "Tech giants pump $235m into AI start-up Hugging Face". In: *Silicon Republic* (2023). URL: https://www.siliconrepublic.com/start-ups/hugging-face-series-d-funding-salesforce-google-amd.

[3] Kyle Rimkus et al. "Digital Preservation File Format Policies of ARL Member Libraries: An Analysis". In: *D-Lib Magazine* 20.3/4 (2014). DOI: 10.1045/march2014-rimkus. URL: https://www.dlib.org/dlib/march14/rimkus/03rimkus.html.

[4] Isabella Peters et al. "Zenodo in the Spotlight of Traditional and New Metrics". In: *Frontiers in Research Metrics and Analytics* 2 (Dec. 2017). DOI: 10.3389/frma.2017.00013.

[5] Irina Bolychevsky. *U.S. government's data portal Data.gov relaunched on CKAN*. URL: https://ckan.org/blog/data-gov-relaunch-on-ckan.

[6] U.S. General Services Administration. *About Us*. URL: https://data.gov/about/.

[7] Lawrence Berkeley National Laboratory. *ESS-DIVE Documentation*. URL: https://docs.ess-dive.lbl.gov/.

[8] Ameriflux. *About Ameriflux*. URL: https://ameriflux.lbl.gov/about/about-ameriflux/.

[9] IEEE. *Welcome to IEEE DataPort*. URL: https://ieee-dataport.org/.

[10] Hugging Face. *Hugging Face Hub Documentation*. URL: https://huggingface.co/docs/hub/en/index.