EMBER2024 - A Benchmark Dataset for Holistic Evaluation of Malware Classifiers

Robert J. Joyce
Booz Allen Hamilton
McLean, VA, USA
Joyce_Robert2@bah.com

Gideon Miller
Laboratory for Physical Sciences
College Park, MD, USA
gmmiller@lps.umd.edu

Phil Roth
CrowdStrike
Austin, TX, USA
phil.roth@crowdstrike.com

Richard Zak
Booz Allen Hamilton
McLean, VA, USA
Zak_Richard@bah.com

Elliott Zaresky-Williams
Booz Allen Hamilton
McLean, VA, USA
Zaresky-Williams_Elliott@bah.com

Hyrum Anderson
Cisco Systems
San Jose, CA, USA
hyrum@cisco.com

Edward Raff
Booz Allen Hamilton
McLean, VA, USA
Raff_Edward@bah.com

James Holt
Laboratory for Physical Sciences
College Park, MD, USA
holt@lps.umd.edu

Abstract

A lack of accessible data has historically restricted malware analysis research, and practitioners have relied heavily on datasets provided by industry sources to advance. Existing public datasets are limited by narrow scope — most include files targeting a single platform, have labels supporting just one type of malware classification task, and make no effort to capture the evasive files that make malware detection difficult in practice. We present EMBER2024, a new dataset that enables holistic evaluation of malware classifiers. Created in collaboration with the authors of EMBER2017 and EMBER2018, the EMBER2024 dataset includes hashes, metadata, feature vectors, and labels for more than 3.2 million files from six file formats. Our dataset supports the training and evaluation of machine learning models on seven malware classification tasks, including malware detection, malware family classification, and malware behavior identification. EMBER2024 is the first to include a collection of malicious files that initially went undetected by a set of antivirus products, creating a "challenge" set to assess classifier performance against evasive malware. This work also introduces EMBER feature version 3, with added support for several new feature types. We are releasing the EMBER2024 dataset to promote reproducibility and empower researchers in the pursuit of new malware research topics.

CCS Concepts

• **Security and privacy → Malware and its mitigation**.

Keywords

Malware, Benchmark Dataset

KDD'25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737431

ACM Reference Format:
Robert J. Joyce, Gideon Miller, Phil Roth, Richard Zak, Elliott Zaresky-Williams, Hyrum Anderson, Edward Raff, and James Holt. 2025. EMBER2024 - A Benchmark Dataset for Holistic Evaluation of Malware Classifiers. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD'25). August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3711896.3737431

1 Introduction

Machine learning is increasingly being applied to several malware classification tasks [21]. Training and evaluating a malware classifier requires a large corpus of recently observed and well-labeled files, but sufficient data is not reasonably accessible to academics [14]. Large security companies use deployed infrastructure and client telemetry to collect malware that is actively being distributed "in the wild" [29]. Sharing agreements between such companies allow them to exchange files and threat intelligence. Commercial feeds of malware exist; however, subscribing to one may not be financially viable for an independent researcher [8, 30]. This restricts access to newly emerging malware, especially if it is needed in large quantities. If a large dataset can be gathered, several challenges in publicly sharing both benign and malicious software often result in it being kept private [9, 18, 23]. These factors have made it difficult for researchers to compare the performance of their own malware classifiers against other work.

To address this reproducibility issue, datasets for benchmarking malware classifiers have been released [2]. However, the most recent large (1M+ file) dataset with both benign and malicious software has files that were collected six years before the time of writing [8]. The ecosystem of malware is constantly changing, and new varieties of malware, malicious techniques, and threat actors are not represented in prior datasets. This malware evolution results in concept drift, and performance degrades when a classifier attempts to detect malware that is newer than its training period [10]. Large datasets with recent malware exist, but they lack benign files, which are necessary for benchmarking malware detection [1, 13]. Other benchmark datasets

Table 1: Notable 1M+ file malware datasets. Our EMBER2024 (E'24) is the only multi-platform dataset to support malware detection, malware family classification, behavior identification. EMBER2024 is also the first to include a "challenge" set and code for replicating our dataset construction methodology.

	SOREL	MalDICT	E'17	E'18	**E'24**
Size	20M	5.5M	1.0M	1.1M	3.2M
Malw. Detection	✓		✓	✓	✓
Fam. Classification				✓	✓
Behav. Prediction	✓	✓			✓
Multi-Platform					✓
Challenge Set					✓
Infrastructure Code					✓

are limited to malware targeting a single platform and/or only have labels supporting a single classification task [2, 8, 22].

1.1 The EMBER2024 Dataset

With 3,238,315 files collected between September 2023 and December 2024, the EMBER2024 dataset provides researchers with a large and representative collection of recent malware. The files in EMBER2024 have **seven types of labels and tags** that support malware detection, family identification, and other relevant multi-label tasks. EMBER2024 is the first malware benchmark to include a **challenge set** of files that were not initially detected by any antivirus products. The dataset includes files in **six file formats** — Win32, Win64, .NET, APK, ELF, and PDF — and we introduce a feature format that allows unified representation and exploration across all formats. To our knowledge, we are also the first to provide code for replicating our dataset construction methodology. These properties aim to make EMBER2024 a holistic malware benchmark capable of assessing malware classifiers in a variety of ways.

The rest of this work is organized as follows. In Section 2 we detail the design choices made in the creation of EMBER2024 and we discuss the process by which a set of antivirus-eluding malware was identified. Section 3 describes the contents of the EMBER2024 dataset and the updates made to the EMBER feature format, including partial support for non-PE files. Several experiments using standard benchmark models are performed in Section 4, challenging conventional wisdom about classifier performance in the presence of evasive malware and concept drift. We present a retrospective written by the authors of EMBER2017 and EMBER2018 in Section 5, describing the impact of these datasets on the malware research community. Finally, we review related work in Section 6 and provide our conclusions in Section 7.

2 Data Collection and Labeling

In this section, we describe the procedure used to build the EMBER2024 dataset. On each day between September 24th, 2023 and December 14th, 2024, we identified a set of files that were first submitted to VirusTotal on that day. For each of those files, we retrieved VirusTotal analysis results (which we refer to as **VirusTotal reports**)


```
{
  "sha256": "b7b78099082384d7da3594121d85dd7f4...",
  "first_submission_date": "2024-01-30T00:00:53",
  "last_analysis_date": "2024-04-06T11:52:27",
  "last_analysis_results": {
    "Microsoft": {
      "category": "malicious"
      "result": "TrojanDownloader:Win32/Nemucod!ml"
    },
    "MaxSecure": {
      "category": "malicious"
      "result": "Trojan.WIN32.cryxos.5913"
    },
    ...
  }
}
```


Figure 1: Example VirusTotal report contents. VirusTotal reports include a file's hash, the date it was first submitted to VirusTotal, and the date the file was most recently analyzed. They also include scan results from ≈70 AV products.

Table 2: EMBER2024 Weekly File Inclusion Thresholds

File Type	Malicious Files	Benign Files	Total
Win32	15,000	15,000	30,000
Win64	5,000	5,000	10,000
.NET	2,500	2,500	5,000
APK	2,000	2,000	4,000
ELF	250	250	500
PDF	500	500	1,000

within 24 hours of first submission. Then, we again queried each of those files 90 or more days after its first submission date.

A VirusTotal report contains information about the queried file, such as its hashes, first submission date, last analysis date, antivirus (AV) detection results, and other metadata. A VirusTotal scan report for a fictitious file is shown in Figure 1. In the 90+ days between queries, files in VirusTotal may have been re-scanned, causing antivirus detections to update. We ensured that all files suspected to be benign received re-scans at least 30 days after first submission, re-scanning them ourselves if necessary. The most recent antivirus detections were then used to label and tag each file, described in more detail in Section 2.4. This methodology ensures that the malware in the EMBER2024 dataset is relevant and accurately labeled.

2.1 File Selection

The file collection period for building the EMBER2024 dataset spans exactly 64 weeks. To encourage research into malware concept drift, EMBER2024 includes an equal number of files per week. Table 2 lists how many files were included in the dataset from each week of data collection, per label and file format. The number of files per file type selected per week was determined by availability. For example, ELF files were the rarest of the six file types in our collection, and this is reflected in EMBER2024.

For each of the 64 weeks of data collection, we gathered all files that we found to have a first VirusTotal submission date within that given week. Files from that week were then bucketed by file type and malicious-benign label. Any files that were not definitively identified

as malicious or benign were discarded. Files were randomly drawn from each bucket (ignoring near-duplicates and files larger than 100MB in size) until the corresponding threshold in Table 2 was reached.

Near-duplicate files were identified using Trend Micro Locality Sensitive Hashing (TLSH) [19]. When considering whether to include a file in the dataset, we compared its TLSH digest against that of each other file already chosen from that week of data collection. If the current file had a TLSH distance of 30 or less to any previously selected file, the pair was considered to be near-duplicates, and the current file was discarded. This threshold was chosen based on the work of Oliver et al. [19], whose evaluation found that a TLSH distance of 30 has a false positive rate of 0.00181% when identifying related files.

2.2 Training and Test Sets

EMBER2024 includes 50,500 files for each of the 64 weeks of data collection, divided into a training and test set. EMBER2024's training set is comprised of files from the first 52 weeks of data collection (Sep. 24, 2023 - Sep. 21, 2024, 2,626,000 files in total), while the final 12 weeks make up the test set (Sep. 22, 2024 - Dec. 14, 2024, 606,000 files in total).

2.3 Challenge Set

The ≈70 AV products that VirusTotal uses for file scanning employ a variety of detection technologies, including file signatures, heuristics, and machine learning. File signatures are used to match known threats, while heuristics and machine learning attempt to detect emerging malware and malware that has changed enough to evade existing signatures [4]. In rare cases, malicious files go fully undetected on VirusTotal until AV products are updated with new signatures. Detecting evasive malware is a research priority, and EMBER2024 is the first dataset to provide a dedicated subset of files for evaluating classifiers on this task.

The 6,315 files in EMBER2024's "challenge" set were not initially detected by any AV products in VirusTotal. However, after being re-scanned at least 30 days later, they were detected by a sufficient number of AV products to receive a malicious label (described in Section 2.4). To maximize the size of the challenge set, files were selected from all 64 weeks of data collection (that is, overlapping the training and test set collection periods). Files in the challenge set do not appear in the EMBER2024 training or test sets. Furthermore, we ensured that for each file in the challenge set, no near-duplicates from the same week of data collection were included in the training or test sets.

Table 3: EMBER2024 Dataset File Statistics

File Type	Train	Test	Challenge	Total
Win32	1,560,000	360,000	3,225	1,923,225
Win64	520,000	120,000	814	640,814
.NET	260,000	60,000	805	320,805
APK	208,000	48,000	256	256,256
ELF	26,000	6,000	386	32,386
PDF	52,000	12,000	805	12,805

2.4 Labeling Methodology

Like similar datasets, our labeling methodology is based on AV detection counts [2]. Files that were not detected by any AV products after being re-scanned at least 30 days after their first submission to VirusTotal are labeled as benign. Files detected as malware by five or more AV products without known relationships (e.g., engine sublicensing, company acquisition, or public data sharing agreements) labeled as malicious [13].

The malware in EMBER2024 is labeled by family using ClarAVy [12]. ClarAVy uses an intelligent Bayesian combination strategy to accurately predict malware families, and each family label has an associated confidence score. Every malicious file is also tagged according to its behaviors, file properties, exploits, packers, and threat group attribution using ClarAVy. Some files may not receive family labels or other tags due to insufficient information in their AV detections, or because no family or tag is applicable. Files may also receive multiple tags within the same category if applicable (for example, a file may have both the "ransomware" and "worm" behavioral tags). File property tags that indicate file format (e.g. "win32", "apk", "pdf") were discarded due to redundancy.

2.5 Family and Tag Demographics

1,356,182 of the 1,616,000 malicious files in the EMBER2024 training and test sets have a malware family label. Consistent with prior work [11], we observe that the malware family sizes in EMBER2024 approximately follow a zipfian distribution. A histogram of malware family sizes is shown in Figure 2. We identified 6,787 unique families among them, and 2,538 of those families have 10 or more instances in EMBER2024. 3,124 families have little representation in EMBER2024, with just five or fewer instances in the dataset. In contrast, there are 12 families that appear more than 10,000 times in EMBER2024. 34.04% of the malicious files in the EMBER2024 training and test sets (550,087 files) belong to one of those 12 families. EMBER2024 has 2,709 files from 75 families that appear 10 or more times in the test set but are not in the training set. In section 4.4 we use this to simulate the emergence of new families over time and evaluate the difficulty in detecting emerging families.

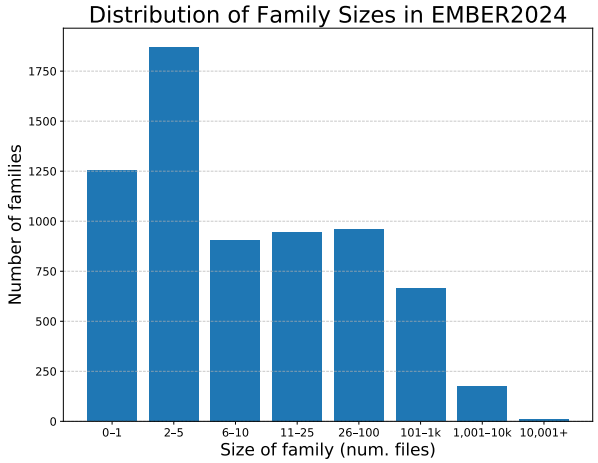

Figure 2: Histogram showing the distribution of family sizes in the EMBER2024 training and test sets.

Table 4: EMBER2024 Tag Statistics

Tag category	# Tagged files	# Total Tags
Behavior	733,142	118
File property	142,199	30
Packer	99,235	52
Exploited vuln.	2,991	293
Threat group	16,170	43

Statistics about tags related to malware behaviors, file properties, packers, exploited vulnerabilities, and threat group attribution are listed in Table 4. Tags that were applied to 10 or more files in EMBER2024 are used to train and evaluate benchmark classifiers in Section 4.6. Few files have tags related to vulnerability exploitation or known threat group.

3 Dataset Contents

The contents of the EMBER2024 dataset include raw file metadata, feature vectors, labels and tags, and trained classifiers. Unfortunately, like EMBER2017 and EMBER2018, we cannot release the original files in EMBER2024. Instead, we are releasing code that allows users with a VirusTotal API key to download these files. We are also publishing other code related to dataset construction, feature extraction, and model training to aid researchers and ensure reproducibility.

3.1 File Metadata and Feature Vectors

The EMBER2024 dataset includes metadata in the form of JSON objects for each file. Files are uniquely identified by MD5, SHA-1, and SHA-256 digests, and TLSH digests enable approximate file comparison. Each JSON object includes UNIX timestamps indicating when a file was first uploaded to VirusTotal and when the file was most recently scanned in VirusTotal (at the time of dataset construction). Each object also lists the ratio of malicious detections to total antivirus scans, the file type, family label, and other tags. Finally, each JSON object includes EMBER feature version 3 raw features for the corresponding file. An JSON object for a fictitious file is shown in Figure 3.

3.2 EMBER Feature Version 3

The EMBER2017 and EMBER2018 datasets established version 1 and version 2 of the EMBER feature vector format, respectively [2, 22]. The EMBER feature format enables researchers to easily obtain raw features and/or feature vectors from Windows Portable Executable (PE) files, and it has been broadly adopted by the malware analysis research community for training classifiers to perform static detection of Windows malware [7, 16, 27, 31]. We introduce EMBER feature version 3, which updates existing feature categories from feature versions 1 and 2 and adds several new feature categories. Furthermore, feature version 3 supports limited feature extraction from non-PE files and PE files that cannot be properly parsed.

EMBER feature version 3 includes features extracted from the PE COFF, Optional, and Section headers that were not included in feature versions 1 or 2. Modifications were made to the set of general file features and more features related to string appearances and statistics have been incorporated. Several new feature categories were added to EMBER feature version 3: DOS header features, data directory features, Rich header features, Authenticode signature features, and PE file parse warning features. Appendix A displays EMBER version 3 raw features extracted from a fictitious file. We discuss each of the feature version 3 changes below.

Additonal PE COFF Header Features. Four features from the PE COFF header were added: The number of PE sections, the size of the PE optional header, the number of symbols, and the pointer to the symbol table.

Additional PE Optional Header Features. 13 features from the PE Optional header were added to EMBER feature version 3. These features are primarily the sizes of, or pointers to, various regions of memory required by the Windows loader when a PE file is executed and becomes a process.

Additional PE Section header features. Feature version 3 reduces the number of features allocated to PE section names. Features such as the ratio of physical section size to total file size and the ratio of physical section size to virtual section size, have been added.

General File Features. The general file feature category has been repurposed, and now includes file size, file entropy, and the first four bytes in the file (for inferring file type). Features specific to Windows PE files have been moved to other feature categories.

String Features. Prior EMBER feature versions included regular expressions that search for strings indicative of file paths, URLs, registry keys, and embedded PE executables. Feature version 3 now searches for 76 strings or string patterns that may be useful for determining a file's behaviors and whether it is malicious or benign.

DOS Header Features. The DOS header remains at the beginning of PE files for legacy purposes. EMBER feature version 3 includes each entry in the DOS header as a feature.

PE Data Directory Features. A PE file may contain several data directories with specialized information, such as debug data,


```
{
  "md5": "93080b69b30c4658ecaf4104f8bf62d5"
  "sha1": "14bb95b5220acb12c328922567cc899330e...",
  "sha256": "b7b78099082384d7da3594121d85dd7f4...",
  "tlsh": "T1002354D8E1FEDE31036602DDB3E9AB5B7...",
  "first_submission_date": 1704706843,
  "last_analysis_date": 1707870639,
  "detection_ratio": "64/75"
  "label": 1,
  "file_type": "Win32",
  "family": "wannacry",
  "family_confidence": 0.961,
  "behavior": ["ransomware", "worm"],
  "file_property": ["packed"],
  "packer": ["upx"],
  "exploit": ["cve-2017-0144],
  "group": [],
  "histogram": [...],
  "byteentropy": [...],
  "strings: {...},
  "general": {...},
  "header": {...},
  "section": {...},
  "imports": {...},
  "exports": [...],
  "datadirectories": {},
  "richheader": [...],
  "authenticode": {...},
  "pefilewarnings": [...],
}
```


Figure 3: Example JSON object displaying a file's hashes, labels, EMBER feature version 3 raw features, and other metadata.

relocation data, and resource data. The names, sizes, and virtual addresses of each data directory are used as features.

Rich Header Features. The Rich header is an undocumented header included in PE files linked using the Windows loader. It includes metadata about artifacts generated during the compilation and linking process. EMBER feature version 3 uses the hashing trick to record each entry in the Rich header.

Authenticode Signature Features. Authenticode is used for Windows PE file code signing. EMBER feature version 3 includes features about Authenticode signatures such as the number of certificates, whether the certificate is self-signed, whether any certificates have empty name values, the date of the most recent certificate, and the difference between this date and the file's compilation timestamp.

PE Parse Warning Features. The `pefile` library (now used for extracting many EMBER raw features) may throw several errors and/or warnings when parsing a file. This is relevant since many malicious PE files — especially those that have been packed or modified after compilation — may not parse correctly. EMBER feature version 3 includes 88 features for tracking various errors and warnings during file parsing.

EMBER feature version 3 is able to partially handle non-PE files and PE files that cannot be parsed. In these instances, the following feature categories can still be extracted: general file features, string patterns and statistics, byte histogram features, and byte entropy histogram features. In Section 4 we show that this limited feature set remains effective for classifying Linux, Android, and PDF malware.

The EMBER feature version 3 raw features vectorize into a feature vector of dimension 2,568 (previously 2,381 under feature version 2). EMBER vectors for non-PE files can be safely truncated to dimension 696, since all further entries in those vectors are zero.

3.3 Source Code

We recognize that, like prior work, the EMBER2024 dataset will become outdated over time. We are publishing code for the following:

- Retrieving VirusTotal reports for a collection of files.
- Computing TLSH digests for a collection of files.
- Labeling a collection of files as malicious or benign using the antivirus results in VirusTotal reports.
- Selecting a preset number of files from a collection, with near-duplicates excluded.
- Downloading selected files from VirusTotal.

This code will allow researchers to replicate the methodology used to compile the EMBER2024 dataset. Note that this requires the use of VirusTotal's API. We are also releasing a Python code implementing the following:

- Extracting EMBER feature version 3 raw metadata.
- Vectorizing EMBER raw features, and writing feature vectors and labels/tags to disk.
- Reading EMBER feature vectors and labels/tags from disk.
- Training `LightGBM` classifiers on EMBER feature vectors.

This released code is an update to the original EMBER Python package. The EMBER feature version 1 and 2 implementations in the original package use the `LIEF` library to extract PE metadata features [28]. However, these implementations are pinned to older versions of `LIEF` and have outdated dependencies. Over time, installing these versions of `LIEF` to compute EMBER raw features has become more burdensome. To rectify this, the implementation of the EMBER feature version 3 implementation has switched from `LEIF` to `pefile`, a well-supported and robust library that has no other dependencies [5].

We have also added code functionality that enables users to load a subset of the EMBER2024 feature vectors for their required classifier training and/or evaluation task(s). Users can easily create custom dataset splits from any combination of the following:

- The training set, test set, or challenge set.
- PE files, Win32 files, Win64 files, .NET files, APK files, ELF files, or PDF files.
- Files with malicious-benign labels (i.e. all files), files with family labels, or files with a certain type of tag.

4 Benchmark Model Results

We are releasing 14 `LightGBM` classifiers trained to perform various malware analysis classification tasks including malware detection, malware family identification, and malware attribute prediction. This section includes training details, benchmark results, and discussion of our findings.

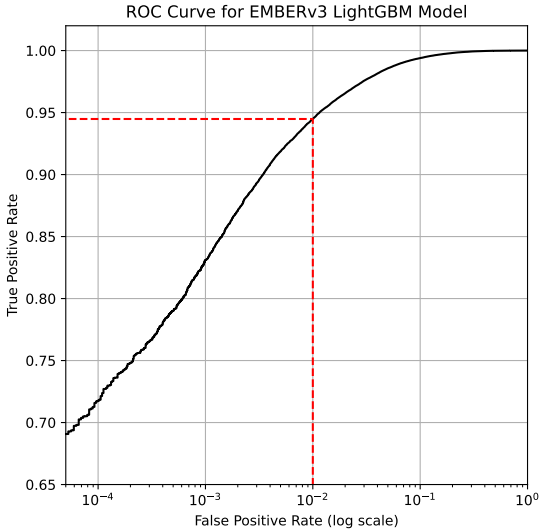

Figure 4: ROC curve (log scale) of a `LightGBM` classifier trained on the EMBER2024 training set and evaluated using the EMBER2024 test set. The model has a true positive rate of 94.48% when permitting a 1% false positive rate.

4.1 Evaluating a Malware Detection Classifier

We trained a `LightGBM` classifier on the EMBER2024 training set for 500 boosting rounds, with 64 leaves and 100 minimum data points required per leaf. These hyper-parameters are identical to the SOREL dataset's benchmark `LightGBM` classifier [8]. Figure 4 shows the classifier's ROC curve for the EMBER2024 test set. The `LightGBM` model has a ROC AUC score of 0.9949 and a true positive rate of 94.48% at 1% false positive rate.

4.2 Evaluating EMBER2024 Subset Classifiers

Eight `LightGBM` classifiers listed in Table 5 were trained on the following partitions of the EMBER2024 training set. The "All files" classifier was trained on the entire EMBER2024 training set, and the "All PE files" classifier was trained using the Win32, Win64, and .NET files in the training set. The "All files" and "All PE files" classifiers were then evaluated several times, using different subsets of the EMBER2024 test set. Classifiers for Win32 files, Win64 files, .NET files, APK files, ELF files, and PDF files were also trained and evaluated using appropriate partitions of EMBER2024. All classifiers use the same hyperparameters described in Section 4.1.

The ROC AUC and Precision-Recall (PR) AUC scores in Table 5 indicate that the trained `LightGBM` classifiers are able to accurately detect malicious files in the EMBER2024 test set, which consists of files that appeared in VirusTotal for the first time 1-12 weeks after the most recent file in the training set. This suggests that the EMBER feature version 3 features are resistant to concept drift due to small temporal changes.

The ROC AUC and PR AUC scores of the ELF and PDF classifiers in Table 5 exceed 0.99, despite being trained on a reduced set of features. The APK classifier was also trained with this limited feature set, and although it was outperformed by the ELF and PDF classifiers, its ROC AUC and PR AUC scores of 0.9868 and 0.9877 demonstrate that it can effectively detect APK malware.

4.3 Evaluations Using the Challenge Set

Next, we repeated the experiments in Section 4.2 using the EMBER2024 challenge set, with results reported in Table 6. Because the challenge set contains only malware, each classifier was evaluated by joining a partition of the challenge set with the benign files in its corresponding partition of the test set. For example, the "PDF files" entry in the "Evaluation partition" column of Table 6 refers to the 805 malicious PDF files in the challenge set plus the 6,000 benign PDF files in the test set.

The ROC AUC scores in Table 6 do not accurately reflect model performance due to class imbalance. PR AUC scores demonstrate that `LightGBM` classifiers struggle to detect the malware in the challenge set, just as AV products initially did. The .NET classifier was able to most accurately identify challenge files, with a PR AUC score of 0.8539. APK files and Win64 files in the challenge set were particularly difficult to detect, with their classifiers having PR AUC scores of 0.2374 and 0.4651 respectively.

Many malware detection tasks require extremely low false positive rates (e.g. below 0.1%), necessitating large evaluation sets to differentiate model performance [8, 20]. The difficulty of the challenge set allows researchers to quickly estimate the relative performance of malware classifiers using a small-scale dataset. Our experiments using the challenge set establish a benchmark for detecting evasive malware and demonstrate that there is potential for significant improvement in this research area.

4.4 Detecting Newly Emerging Families

Recall that the EMBER2024 test set includes 2,709 files from 75 families of size 10 or greater that do not appear in the training set. These files simulate "newly emerging" families for which signatures may not yet exist, and prior work indicates that these files may be more difficult to detect [20]. We combined these 2,709 malicious files with the 303,000 benign files in the EMBER2024 test set and evaluated the `LightGBM` malware detection classifier from Section 4.1 using them. The classifier's resulting PR AUC score was 0.8992.

Table 5: ROC AUC and PR AUC scores for `LightGBM` classifiers trained on various partitions of the EMBER2024 training set and evaluated using different parts of the EMBER2024 test set.

Training partition	Evaluation partition	ROC AUC	PR AUC
All files	All files	0.9969	0.9971
	All PE files	0.9982	0.9983
	Win32 files	0.9981	0.9983
	Win64 files	0.9983	0.9985
	.NET files	0.9968	0.9968
	APK Files	0.9726	0.9737
	ELF files	0.9887	0.9902
	PDF files	0.9878	0.9901
All PE files	All PE files	0.9982	0.9983
	Win32 files	0.9982	0.9984
	Win64 files	0.9986	0.9987
	.NET files	0.9971	0.9971
Win32 files	Win32 files	0.9984	0.9986
Win64 files	Win64 files	0.9989	0.9990
.NET files	.NET files	0.9980	0.9981
APK files	APK files	0.9868	0.9877
ELF files	ELF files	0.9933	0.9933
PDF files	PDF files	0.9912	0.9933

Table 6: ROC AUC and PR AUC scores for `LightGBM` classifiers, using malicious files in the EMBER2024 challenge set plus benign files in the EMBER2024 test set.

Training partition	Evaluation partition	ROC AUC	PR AUC
All files	All files	0.9542	0.5722
	All PE files	0.9643	0.6250
	Win32 files	0.9662	0.6526
	Win64 files	0.9424	0.3214
	.NET files	0.9773	0.7940
	APK files	0.8700	0.7644
	ELF files	0.8975	0.5200
	PDF files	0.7804	0.6145
All PE files	All PE files	0.9661	0.6354
	Win32 files	0.9675	0.6540
	Win64 files	0.9464	0.3509
	.NET files	0.9790	0.8066
Win32 files	Win32 files	0.9689	0.6646
Win64 files	Win64 files	0.9503	0.4651
.NET files	.NET files	0.9865	0.8539
APK files	APK files	0.9101	0.2374
ELF files	ELF files	0.9311	0.6008
PDF files	PDF files	0.9066	0.7841

Our findings in Section 4.2 imply that our baseline LightGBM classifiers are resistant to multiple weeks of concept drift. During such a period, new versions of existing families are introduced and entirely novel families emerge. It seems that our benchmark LightGBM classifiers are well-equipped to identify derivative versions of existing families. However, performance clearly degrades for detection of novel families, and we encourage more study on this topic.

4.5 Evaluating Malware Family Classification

Next, we trained a LightGBM classifier to perform malware family identification. We identified 2,358 families that appear 10 or more times in the EMBER2024 training set. Files not in one of these 2,358 families were not used for model training. The remaining files were divided using a stratified split, with 90% of each family used for training and 10% for validation during training. The LightGBM classifier was trained for 100 boosting rounds, with 64 leaves and 10 minimum data points per leaf. Early stopping was permitted after 10 boosting rounds. Evaluation was performed using all files with family labels in the EMBER2024 test set, and the classifier achieved an accuracy of 67.97%. We also computed the precision, recall, and F1 score of the classifier using both macro averaging and weighed averaging, and these results are shown in Table 7.

The model's performance metrics are markedly lower when using macro averaging. This suggests that the model performs well in detecting common families, but has more difficulty classifying the (many) smaller families in the dataset. We believe that this is primarily due to lack of sufficient training data for these smaller families.

4.6 Multi-Label Malware Classification Tasks

LightGBM classifiers were trained to perform the following multi-label tasks: behavior prediction, file property prediction, packer identification, exploited vulnerability identification, and threat group identification. Classifiers were trained on individual tags using a One-Vs-Rest (OvR) approach, using the same hyper-parameters as the LightGBM model in Section 4.5. Tags that occurred fewer than 10 times in the EMBER2024 training set were discarded from the training and test sets.

The results in Table 8 show that the classifiers clearly struggled to generalize in all five of these tasks, with low results across all metrics. The precision of models with a large number of tags was especially low. Like family classification, the poor macro-averaging results are likely due to limited training data for many tags, in addition to the difficulty of each task. Prior work also points to a temporal train/test split contributing to lowered performance in multi-label malware classification [13].

4.7 Discussion of Benchmark Results

The LightGBM classifiers used in our experiments have not been tuned and are meant to leave room for optimization. Improvements in hyper-parameter selection, model choice, and training strategy will likely yield better performance. Rather, these models are meant to serve as benchmarks that demonstrate the results that can be expected from a basic classifier trained on these tasks. EMBER2024 enables researchers to publish reproducible results by evaluating their classifiers against our benchmark models and models trained by others.

Table 7: Family classification results for a LightGBM model trained to identify 2,358 families.

Metric	Score (macro avg.)	Score (weighted avg.)
Precision	0.5670	0.7360
Recall	0.3980	0.6797
F1 score	0.4371	0.6664

Table 8: Precision, Recall, F1 Measure, and average AUC (using macro averaging) of One-vs-Rest (OvR) LightGBM classifiers trained to predict tags in EMBER2024.

Pred. Task	# Tags	Precision	Recall	F1	AUC
Behavior	92	0.0981	0.5254	0.1345	0.7558
File property	20	0.3037	0.5328	0.3451	0.7462
Packer	32	0.2066	0.6722	0.2525	0.8310
Exploited vuln.	46	0.5038	0.6570	0.5102	0.8192
Threat group	6	0.7588	0.5488	0.5823	0.7737

The LightGBM benchmark classifiers of EMBER2018 and SOREL-20M have ROC AUC scores of 0.996 and 0.998, respectively, despite attempts to include more "difficult" malware than EMBER2017 [2, 8, 22]. However, our studies on evasive malware and malware from novel families show that malware detection is far from a solved problem. Despite our own LightGBM classifier having a ROC AUC score (0.9969) similar to past benchmarks, we identified populations of malicious files that can reliably bypass detection. We believe that further studies in this area are warranted, and the EMBER2024 test and challenge sets will support this research.

5 EMBER Dataset Retrospective

Previous to the release of the first EMBER dataset and accompanying LightGBM model, several pioneering works applied machine learning to train malware classifiers [6, 15, 23, 24]; however, datasets were proprietary and/or very small (a few thousand samples). Initially released under the generous support of Endgame (now part of Elastic) in 2018, EMBER was created with a straightforward goal: to provide a standardized benchmark dataset to "invigorate machine learning research for malware detection" in much the same way that benchmark datasets had done for computer vision [2]. We considered a number of research use-cases that included baselining malware classification performance with the co-released LightGBM model, adversarial machine learning offense and defense, semi-supervised learning for malware detection, among others.

Since its release, the original EMBER dataset has been cited over 600 times from more than 350 unique citing institutions across 6 continents, in what we considered to be a relatively niche research field at the time. A sampling of papers shows an 82% / 18% split between academia / industry affiliations. A brief survey of citing publications indicate that the EMBER publication has also spurred the release of other malware datasets, in addition to innovations in defensive ML security (e.g., malware classification), offensive ML security

(e.g., malware evasion), and advancements in ML architectures or algorithms.

Table 9: Topics of papers that cited the first EMBER dataset, as adjudicated with the assistance of GPT-4o.

Category	Percent
Defensive ML Security	36.2%
Survey Papers	19.8%
Other Benchmark Datasets	19.0%
Offensive ML Security	18.1%
ML Architecture or Algorithms	6.9%

Besides academic publications, a host of unpublished work from malware offensive and defensive competitions [3, 17] has engaged security practitioners. Email interactions with educators indicate that the dataset and model are being used at institutions that range from high school to graduate school (and no, we are still legally unable to provide benign files).

In summary, we have been overwhelmed by the response to the EMBER dataset. This overdue update will make the EMBER features easier to calculate, includes more capable features for Windows PE files, expands support beyond Windows PE files, and will enable yet another generation of researchers to advance the state of the art for applying machine learning to malware detection and related challenges.

6 Related Work

Following EMBER2017 and EMBER2018, other 1M+ file malware datasets have contributed to the malware research domain. The SOREL-20M dataset was the first large, labeled dataset to release disarmed malicious executables [8], and it is currently the largest dataset with labeled malicious and benign files at the time of writing. Furthermore, the malware in SOREL-20M is tagged according to 11 behavioral properties. MalDICT is a malware-only dataset tagged according to malicious behaviors, file properties, exploited vulnerabilities, and file packers [13]. It made benchmarking less common malware classification tasks possible for the first time. The VirusShare collection is the largest public malware corpus to our knowledge, with 41,680,896 files available for download at the time of writing. AVClass labels for ≈79.5% of these files (from April 2019 and earlier) are available online [25, 26]. VirusShare is regularly updated with new malware, but does not include benign files.

7 Conclusion

The malware research community is long overdue for another large malware benchmark dataset. EMBER2024 gives researchers access to metadata, feature vectors, and labels for more than 3.2 million malicious and benign files. Including six file formats and seven types of labels and tags, our dataset makes holistic evaluation of malware classifiers attainable. To our knowledge, the EMBER2024 challenge set is the first of its kind, enabling new studies on evasive malware. As a result of our inquiries into evasive malware and newly emerging families, we advocate for further study on developing robust malware classifiers. We also made several other contributions in this work, such as code for replicating our dataset building methodology, an updated EMBER version 3 feature vector format, and 14 trained benchmark classifiers. The main EMBER2024 GitHub repository can be found at https://github.com/FutureComputing4AI/EMBER2024. Code for replicating our dataset building methodology is located at https://github.com/FutureComputing4AI/vtpipeline-rs. It is our hope that EMBER2024 will become a valuable resource for researchers and a catalyst for investigating critical malware analysis topics.

References

[1] [n. d.]. VirusShare.com - Because Sharing is Caring. https://virusshare.com/, Last accessed on 2025-02-17.

[2] Hyrum S Anderson and Phil Roth. 2018. Ember: an open dataset for training static pe malware machine learning models. *arXiv preprint arXiv:1804.04637* (2018).

[3] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy. 2023. "Real attackers don't compute gradients": Bridging the gap between adversarial ML research and practice. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 339–364.

[4] Marcus Botacin, Fabricio Ceschin, Paulo De Geus, and André Grégio. 2020. We need to talk about antiviruses: challenges & pitfalls of av evaluations. *Computers & Security* 95 (2020), 101859.

[5] Ero Carrera. 2004. pefile: Python module to read and work with PE files. https://github.com/erocarrera/pefile

[6] William W Cohen. 1995. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*. 115–123.

[7] Colin Galen and Robert Steele. 2020. Evaluating performance maintenance and deterioration over time of machine learning-based malware detection models on the ember pe dataset. In *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 1–7.

[8] Richard Harang and Ethan M Rudd. 2020. SOREL-20M: A large scale benchmark dataset for malicious PE detection. *arXiv preprint arXiv:2012.07634* (2020).

[9] Wenyi Huang and Jack W Stokes. 2016. MtNet: a multi-task neural network for dynamic malware classification. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 13th International Conference, DIMVA 2016, San Sebastián, Spain, July 7-8, 2016, Proceedings 13*. Springer, 399–418.

[10] Roberto Jordaney, Kumar Sharad, Santanu K. Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. 2017. Transcend: Detecting Concept Drift in Malware Classification Models. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 625–642. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/jordaney

[11] Robert J Joyce, Dev Amlani, Charles Nicholas, and Edward Raff. 2023. Motif: A malware reference dataset with ground truth family labels. *Computers & Security* 124 (2023), 102921.

[12] Robert J. Joyce, Derek Everett, Maya Fuchs, Edward Raff, and James Holt. 2025. ClarAVy: A Tool for Scalable and Accurate Malware Family Labeling. In *Companion of the 16th ACM/SPEC International Conference on Performance Engineering (WWW Companion '25)*.

[13] Robert J Joyce, Edward Raff, Charles Nicholas, and James Holt. 2023. MalDICT: Benchmark Datasets on Malware Behaviors, Platforms, Exploitation, and Packers. *arXiv preprint arXiv:2310.11706* (2023).

[14] Alex Kantchelian, Michael Carl Tschantz, Sadia Afroz, Brad Miller, Vaishaal Shankar, Rekha Bachwani, Anthony D. Joseph, and J. D. Tygar. 2015. Better Malware Ground Truth: Techniques for Weighting Anti-Virus Vendor Labels. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security* (Denver, Colorado, USA) *(AISec '15)*. Association for Computing Machinery, New York, NY, USA, 45–56. doi:10.1145/2808769.2808780

[15] Jeremy Z Kolter and Marcus A Maloof. 2004. Learning to detect malicious executables in the wild. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 470–478.

[16] Nicola Loi, Claudio Borile, and Daniele Ucci. 2021. Towards an automated pipeline for detecting and classifying malware through machine learning. *arXiv preprint arXiv:2106.05625* (2021).

[17] Microsoft Security Team. 2021. *Attack AI Systems in Machine Learning Evasion Competition*. https://www.microsoft.com/en-us/security/blog/2021/07/29/attack-ai-systems-in-machine-learning-evasion-competition/

[18] Aziz Mohaisen, Omar Alrawi, and Manar Mohaisen. 2015. AMAL: High-fidelity, behavior-based automated malware analysis and classification. *Computers & Security* 52 (2015), 251 – 266. doi:10.1016/j.cose.2015.04.001

[19] Jonathan Oliver, Chun Cheng, and Yanggui Chen. 2013. TLSH–a locality sensitive hash. In *2013 Fourth Cybercrime and Trustworthy Computing Workshop*. IEEE, 7–13.

[20] Tirth Patel, Fred Lu, Edward Raff, Charles Nicholas, Cynthia Matuszek, and James Holt. 2023. Small Effect Sizes in Malware Detection? Make Harder Train/Test Splits! (2023).

[21] Edward Raff and Charles Nicholas. 2020. A survey of machine learning methods and challenges for windows malware classification. *arXiv preprint arXiv:2006.09271* (2020).

[22] Phil Roth. 2019. EMBER Improvements. (2019). https://docs.google.com/presentation/d/1A13tsUkgWeujTy9SD-vDFfQp9fnIqbSE_tCihNPlArQ Conference on Applied Machine Learning in Information Security.

[23] Joshua Saxe and Konstantin Berlin. 2015. Deep neural network based malware detection using two dimensional binary program features. In *Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on*. IEEE, 11–20.

[24] Matthew G Schultz, Eleazar Eskin, F Zadok, and Salvatore J Stolfo. 2001. Data mining methods for detection of new malicious executables. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*. IEEE, 38–49.

[25] Marcos Sebastián, Richard Rivera, Platon Kotzias, and Juan Caballero. 2016. Avclass: A tool for massive malware labeling. In *Research in Attacks, Intrusions, and Defenses: 19th International Symposium, RAID 2016, Paris, France, September 19-21, 2016, Proceedings 19*. Springer, 230–253.

[26] John Seymour. [n. d.]. label-virusshare. https://github.com/seymour1/label-virusshare, Last accessed on 2025-02-14.

[27] Wei Song, Xuezixiang Li, Sadia Afroz, Deepali Garg, Dmitry Kuznetsov, and Heng Yin. 2021. MAB-Malware: A Reinforcement Learning Framework for Attacking Static Malware Classifiers. arXiv:2003.03100 [cs.CR] https://arxiv.org/abs/2003.03100

[28] Romain Thomas. 2017. LIEF - Library to Instrument Executable Formats. https://lief.quarkslab.com/.

[29] Xabier Ugarte-Pedrero, Mariano Graziano, and Davide Balzarotti. 2019. A close look at a daily dataset of malware samples. *ACM Transactions on Privacy and Security (TOPS)* 22, 1 (2019), 1–30.

[30] VirusTotal. [n. d.]. Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community. https://www.virustotal.com/en/, Last accessed on 2025-02-18.

[31] Shao-Huai Zhang, Cheng-Chung Kuo, and Chu-Sing Yang. 2019. Static PE Malware Type Classification Using Machine Learning Techniques. In *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*. 81–86. doi:10.1109/ICEA.2019.8858297

A EMBER Feature Version 3 Raw Features


```
{
  "histogram": [67647, 42400, 37862, ..., 32387, 33015,
  ↪  37394],
  "byteentropy": [0, 0, 0, 0, ..., 1058323, 1063221,
  ↪  1051062],
  "strings": {
    "numstrings": 43473,
    "avlength": 6.213189795965312,
    "printabledist": [3321, 2643, 2944, 2692, 3016, ...],
    "printables": 270106,
    "entropy": 6.582472077598922,
    "string_counts": {
      "btc_wallet": 1,
      "certificate": 8,
      "connect": 11,
      "crypt": 31
      ...
    }
  },
  "general": {
    "size": 8782336,
    "vsize": 8880128,
    "has_relocs": 1,
    "has_dynamic_relocs": 0,
    "symbols": 0
  },
  "header": {
    "coff": {
      "timestamp": 1695592800,
      "machine": "IMAGE_FILE_MACHINE_AMD64",
      "number_of_sections": 12,
      "number_of_symbols": 0,
      "sizeof_optional_header": 240,
      "pointer_to_symbol_table": 0,
      "characteristics": ["EXECUTABLE_IMAGE", ...],
    },
    "optional": {
      "magic": 523,
      "subsystem": "IMAGE_SUBSYSTEM_WINDOWS_CUI",
      "major_image_version": 0,
      "minor_image_version": 0,
      "major_linker_version": 2,
      "minor_linker_version": 38,
      "major_operating_system_version": 4,
      "minor_operating_system_version": 0,
      "major_subsystem_version": 5,
      "minor_subsystem_version": 2,
      "sizeof_code": 115200,
      "sizeof_headers": 1024,
      "sizeof_image": 8880128,
      "sizeof_initialized_data": 8781312,
      "sizeof_uninitialized_data": 65024,
      "sizeof_stack_reserve": 2097152,
      "sizeof_stack_commit": 4096,
      "sizeof_heap_reserve": 1048576,
      "sizeof_heap_commit": 4096,
      "address_of_entrypoint": 4389,
      "base_of_code": 4096,
      "base_of_data": 0,
      "image_base": 5368709120,
      "section_alignment": 4096,
      "checksum": 184607,
      "number_of_rvas_and_sizes": 16,
      "dll_characteristics": ["HIGH_ENTROPY_VA", ...],
    },
    "dos": {
      "e_magic": 23117,
      "e_cblp": 144,
      "e_cp": 3,
      "e_crlc": 0,
      "e_cparhdr": 4,
      ...
      "e_minalloc": 0,
      "e_maxalloc": 65535,
      "e_ss": 0,
      "e_sp": 184,
      "e_csum": 0,
      "e_ip": 0,
      "e_cs": 0,
      "e_lfarlc": 64,
      "e_ovno": 0,
      "e_oemid": 0,
      "e_oeminfo": 0,
      "e_lfanew": 128
    }
  },
  "section": {
    "entry": ".text",
    "sections": [
      {
        "name": ".text",
        "size": 115200,
        "entropy": 6.2928493046865155,
        "vsize": 115080,
        "size_ratio": 0.013117238966944557,
        "vsize_ratio": 1.0010427528675705,
        "props": ["CNT_CODE", ...],
      },
      ...
    ],
    "overlay": {
      "size": 6935696,
      "size_ratio": 0.9556944024939537,
      "entropy": 7.997956389085634
    }
  },
  "imports": {
    "KERNEL32.dll": ["CloseHandle", "CopyFileW", ...],
    "SHELL32.dll": ["SHFileOperationW", ...],
    ...
  },
  "exports": [],
  "datadirectories": [
    {
      "name": "RESOURCE",
      "size": 8643008,
      "virtual_address": 229376
    },
    {
      "name": "IAT",
      "size": 768,
      "virtual_address": 217936
    },
    ...
  ],
  "richheader": [1704619, 7, 17135691, 191, 170705, ...],
  "authenticode": {
    "num_certs": 2,
    "self_signed": 1,
    "empty_program_name": 0,
    "no_countersigner": 0,
    "parse_error": 0,
    "chain_max_depth": 7,
    "latest_signing_time": 1643104921,
    "signing_time_diff": 19080976
  },
  "pefilewarnings": [
    "RVA AddressOfFunctions in the export directory...",
    "Invalid IMAGE_DYNAMIC_RELOCATION_TABLE...",
    ...
  ]
}
```


Figure 5: Example of EMBER feature version 3 raw features.

B Top Families and Tags in EMBER2024

Top Families	
berbew	174,481
wacatac	81,556
expiro	74,340
cosmu	53,965
xmrig	28,904
upatre	25,296
sfone	22,177
glupteba	21,670
grandoreiro	20,551
flystudio	18,141

Top Behaviors	
backdoor	228,363
virus	121,971
worm	76,115
downloader	61,625
spyware	55,839
coinminer	37,688
dropper	33,308
adware	24,902
phishing	21,913
ransom	16,295

Top Packers	
upx	31,101
vmprotect	28,280
themida	20,509
nsis	10,952
enigmaprotector	2,472
petite	1,382
mpress	678
nspm	626
obsidium	596
aspack	418

Top File Properties	
msil	100,401
vb	19,734
python	9,749
codecpack	4,499
autoit	3,069
bat	1,358
hllo	1,066
js	961
shellcode	859
powershell	479

Top Exploited Vulns	
cve_2017_17215	926
cve_2017_0147	290
ms17_010	143
cve_2007_5659	114
cve_2020_0601	87
cve_2017_172	79
cve_2015_0057	66
cve_2010_2883	64
cve_2014_8361	59
cve_2018_4993	57

Top Threat Groups	
gamaredon	14,978
turla	475
equationgroup	213
molerats	110
apt28	76
knotweed	53
donotteam	30
lazarusgroup	29
darkhotel	28
apt29	23

Figure 6: Most common families and tags in the EMBER2024 dataset.