MalGrid: Visualization Of Binary Features In Large Malware Corpora

Tajuddin Manhar Mohammed
Mayachitra, Inc.
Santa Barbara, California
mohammed@mayachitra.com

Lakshmanan Nataraj*
Mayachitra, Inc.
Santa Barbara, California
lakshmanan_nataraj@ece.ucsb.edu

Satish Chikkagoudar
U.S. Naval Research Laboratory
Washington, D.C.
satish.chikkagoudar@nrl.navy.mil

Shivkumar Chandrasekaran
Mayachitra, Inc.
ECE Department, UC Santa Barbara
Santa Barbara, California
shiv@ucsb.edu

B.S. Manjunath
Mayachitra, Inc.
ECE Department, UC Santa Barbara
Santa Barbara, California
manj@ucsb.edu

***Abstract*—The number of malware is constantly on the rise. Though most new malware are modifications of existing ones, their sheer number is quite overwhelming. In this paper, we present a novel system to visualize and map millions of malware to points in a 2-dimensional (2D) spatial grid. This enables visualizing relationships within large malware datasets that can be used to develop triage solutions to screen different malware rapidly and provide situational awareness. Our approach links two visualizations within an interactive display. Our first view is a spatial point-based visualization of similarity among the samples based on a reduced dimensional projection of binary feature representations of malware. Our second spatial grid-based view provides a better insight into similarities and differences between selected malware samples in terms of the binary-based visual representations they share. We also provide a case study where the effect of packing on the malware data is correlated with the complexity of the packing algorithm.**

***Index Terms*—Cyber Security, Malware, Data Visualization, Machine Learning, Packing**

I. Introduction

A large number of malware and exploits frequently target military networked computing systems. In a recent report, Antivirus software vendor Kaspersky Lab reported that they processed 380,000 malware on average per day [1]. Though most new malware are modifications of existing ones, their sheer number is quite overwhelming.

Though there are lots of methods to detect malware and their variants, it will be helpful if the millions of malware can be mapped and spatially visualized on a 2D spatial grid. The advantages of spatial visualization are many fold:

1) It is simple to "semantically" arrange malware on the grid using different distance measures.
2) The spatial grid can hold empty spots for potentially future/unseen malware.

Acknowledgement: This work was supported by the ONR contract #N68335-17-C-0048. The views expressed in this paper are the opinions of the authors and do not represent official positions of the Department of the Navy. *Lakshmanan Nataraj contributed to this work while he was employed at Mayachitra.

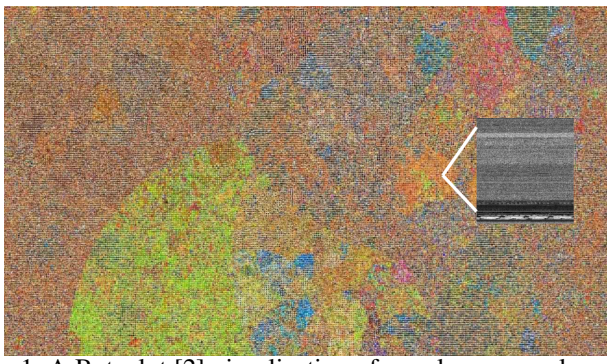


Fig. 1: A Byteplot [2] visualization of a malware sample overlaid on the 80-million tiny images poster [3] (demonstration purpose only).

3) Malware variants get easily grouped as a "cloud" inside the grid.
4) Different visualizations can be produced by varying the feature/distance measure, which would give new insights on the overall "malware space".

The existence of robust feature representations of malware binaries in the literature enables such spatial visualization of the data. Our inspiration to spatially visualize malware comes from the 80-Million tiny images visual dictionary of words and their images, overlaid on a spatial grid [3]. Fig. 1 shows an example of how a malware sample (here, a Byteplot [2] image representation) can be overlaid on the 2D spatial grid of 80-million tiny images poster (shown for demonstration purpose). Here, thousands of images are overlaid on the spatial grid-based on textual and visual similarity. Since the number of malware is also high, the binaries can be visualized as points or thumbnails according to their similarity. In this paper, we propose two such visualizations of malware and benign samples within an interactive display. Our first view is a spatial point-based visualization of similarity among the samples based on a reduced dimensional projection of binary feature representations. Our second visualization is a spatial grid-

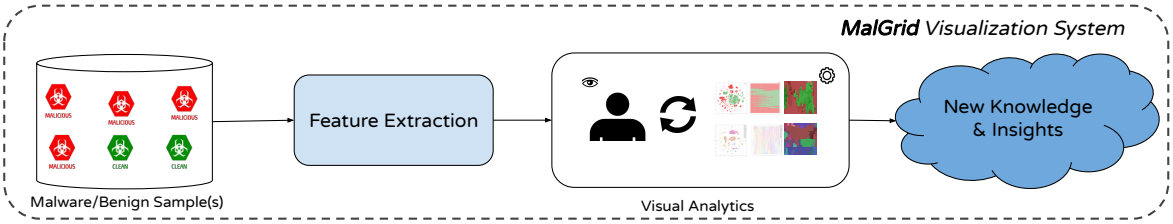


Fig. 2: Block schematic of *MalGrid* visualization system to visualize large malware corpora.

based view which provides a better insight into similarities and differences between selected malware samples in terms of the binary-based visual representations they share. We also provide a case study where the effect of packing on the malware data is correlated with the complexity of the packing algorithm. A block schematic of our *MalGrid* visualization system is shown in Fig. 2.

The main contributions in this paper are as follows:

- First, we propose a method to spatially visualize malware as points based on their similarity.
- Second, we propose a method to visualize malware thumbnails on a 2D grid-based on visual similarity features.
- Finally, we provide practical case studies where our visualizations can be applied in real-life scenarios.

The rest of the paper is organized as follows. In Sec. II, we discuss the related work in malware classification and visualization. In Sec. III and Sec. IV, we look at two different but related malware corpora visualization techniques which also includes a case study for the effect of various packing techniques on the visualization methods. Finally, we conclude in Sec. V. The high-quality images in this paper can be found on Github (https://github.com/Mayachitra-Inc/MalGrid).

II. Related Work

Though there are several methods to detect and classify malware, there are few works that focus on visualization of malware. [4] provides a survey of visualization systems for malware analysis that looks at some of these static features that are effective for visualizing similarities among the malware variants. Most malware visualization methods focus on visualizing a single malware. In [5], self organizing maps are used to detect and visualize malicious code inside an executable. A reverse engineering-based visualization framework is proposed in [6]. In [7], the distributions of operations within a malware are visualized using treemaps, while in [8], a visual analysis environment is proposed that can aid software developers to better understand the code. All the above works focus primarily on visualizing a single malware, and few works [9]–[14] focus on visualizing malware datasets, but uses complex and non-scalable feature representations of malware binaries. Of these, the method proposed in [11] is closest to our work, where grid-based representations of malware samples are shown. However, their approach is more complex since it computes features based on system call behavior logs, which requires the malware sample to be executed. In contrast, our approach is based on visual similarity features which does not require the malware to be executed. Also, the effect of packing on the visualizations have not been extensively studied in these works. For the visual similarity features, we have considered both the hand-crafted features and deep learning-based features. For hand-crafted features, we have used the well-known Byteplot GIST descriptors [2] that has been extensively shown to be effective for the malware classification task in various works [2], [15]–[17]. For deep learning-based features, we have used the ImageNet [18] pre-trained features that has shown to perform well for image-based malware classification via transfer learning [19], [20].

III. Spatial Point Based Visualization of Malware

Dimensionality reduction is one of the most popular methods to visualize high-dimension data in low-dimensional space. t-Distributed Stochastic Neighbor Embedding (t-SNE) [21] is one such commonly used technique to visualize such data in 2D/3D. As a preliminary experiment to see how effectively malware samples can be visualized in lower dimensional space, we randomly choose a few thousand samples from VirusShare [22]. Then, we scan these samples using VirusTotal [23] and based on number of VirusTotal positive hits, we label these samples as either benign (0%), or malware (100%), or unknown (<100%). We further prune these samples to 15,000 (5,000 samples in each category). We then convert these samples to grayscale Byteplot images following the approach of [2] and then compute GIST image features [24] (320-dimensional) on these images.

Fig. 3 shows t-SNE visualization of Image-based Byteplot GIST [2] features for the 15,000 sample set. Preliminary analysis on the samples belonging to each of the larger (malicious) "clusters" indicate that the files corresponding to the samples in a cluster have similar file sizes and similar Byteplot images.

Other visualization methods

Further, we also considered other dimensionality reduction techniques for malware visualizations including UMAP (Uniform Manifold Approximation And Projection) [25], PCA (Principal Component Analysis) [26], and Random Projection [27]. Fig. 4 shows the visualization of the same GIST features corresponding to the 15,000 sample set using various techniques. From the figure, it looks like (atleast, by using the default parameters in the visualization functions) t-SNE

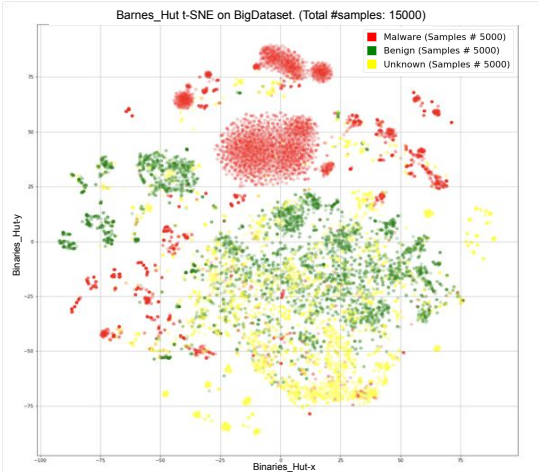


Fig. 3: t-SNE visualization of Byteplot GIST [2] (320-dimensional) features for 15,000 samples – 5,000 malware (RED), 5,000 benign (GREEN), and 5,000 unknown (YELLOW).

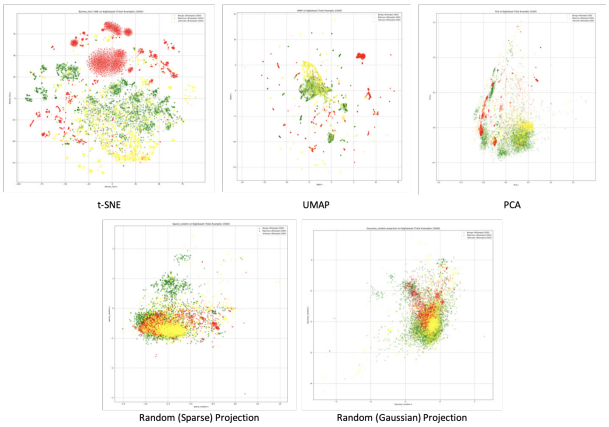


Fig. 4: Visualization of Byteplot GIST [2] (320-dimensional) features for 15,000 samples using different dimensionality reduction techniques.

is better than the others in spreading out and clustering the samples.

A. Point-based Visualization

As t-SNE is proficient in forming the clusters, one can correlate the malware classification performance numbers with the formation of such clusters. As the clusters get more and more distinguishable in the point cloud, the performance in detecting and classifying samples from those cluster-representing malware classes gets better. Since t-SNE is used to visualize the features as a 2D point cloud, the "circular" shape of the plot can be transformed into "rectangular" raster grid while preserving the neighborhood relations that are present in the original cloud. As a result, we have two variations of point based visualizations – (1) 2D t-SNE, and (2) 2D Grid.

For the following discussions, we have used both these representations while visualizing the malware database samples. We have selected the standard object detection model architectures trained on ImageNet and considered the features obtained from the last fully-connected dense layer of the network to obtain the "Pre-ImageNet" features. The architectures include VGG-16 [28], Xception [29] and ResNet-152 [30]. The feature dimensional lengths from these architectures are 4096, 2048 and 2048 respectively. We also considered the aforementioned Byteplot GIST descriptors [2]. We report 10-fold cross validation classification accuracies to support the point-based visualizations that follow. In addition to accuracies, we also report the macro-precision, macro-recall and macro-f1 scores for completeness. The classifiers used are K-Nearest Neighbors (KNN) and Random Forests (RF).

1) Visualization of Malware and Benign samples

Smalldata [31] is a curated dataset that was created by collecting malicious samples from *Malimg* [2], and scraping benign samples from Windows system files. The collection has 9,339 malware and 7,228 benign samples. Table I shows

TABLE I: 10-fold cross-validation binary classification performance for different features and "best" classifiers for *Smalldata* [31] dataset – #malware samples: 9,339, #benign samples: 7,228.

Feature ← <model>	Accuracy	Precision	Recall	F1
Byteplot GIST ← RF	0.970	0.968	0.972	0.970
Pre-ImageNet (VGG16) ← RF	**0.993**	**0.993**	**0.993**	**0.993**
Pre-ImageNet (Xception) ← KNN	0.979	0.981	0.978	0.979
Pre-ImageNet (ResNet152) ← RF	0.990	0.989	0.991	0.990

the binary classification performance using hand-crafted GIST and different deep learning-based features on the dataset. Overall, the "best" feature and classifier are Pretrained-ImageNet (VGG16) and Random Forest (RF) respectively with accuracy, macro-precision, macro-recall and macro-f1 scores of `0.993` each. We present the point-based visualizations corresponding to this feature and model in Fig. 5. From the figure, one can see that the benign and malware samples are distinguishable in both the point cloud and the rectangular grid. These visualizations with groupings of samples not only show the efficacy of our approach to visualize malware and benign samples in a grid-like fashion that contain distinguishable parts, but also support the performance numbers that we get from the binary classification experiments.

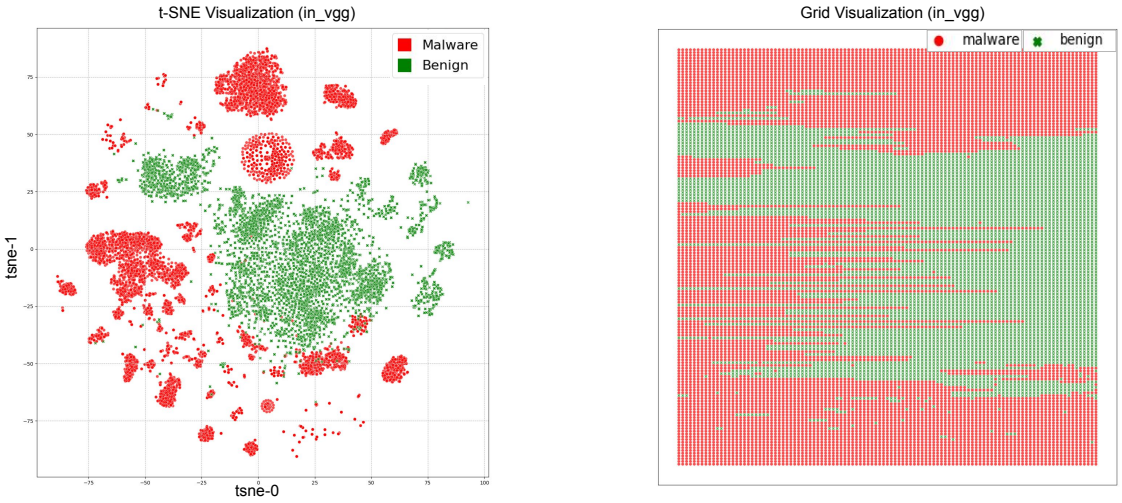
2) Visualization of Malware families

Malimg [2] is a well-known malware image dataset in the literature that has images corresponding to 9,339 malware executables categorized in 25 different malware classes. Table II shows the binary classification performance using

TABLE II: 10-fold cross-validation multi-class classification performance for different features and "best" classifiers for *Malimg* [2] dataset – #samples: 9,339, #malware families: 25.

Feature ← <model>	Accuracy	Precision	Recall	F1
Byteplot GIST ← RF	0.974	0.942	0.933	0.933
Pre-ImageNet (VGG16) ← RF	0.984	0.961	0.957	0.958
Pre-ImageNet (Xception) ← KNN	0.964	0.933	0.906	0.909
Pre-ImageNet (ResNet152) ← RF	**0.987**	**0.970**	**0.965**	**0.966**

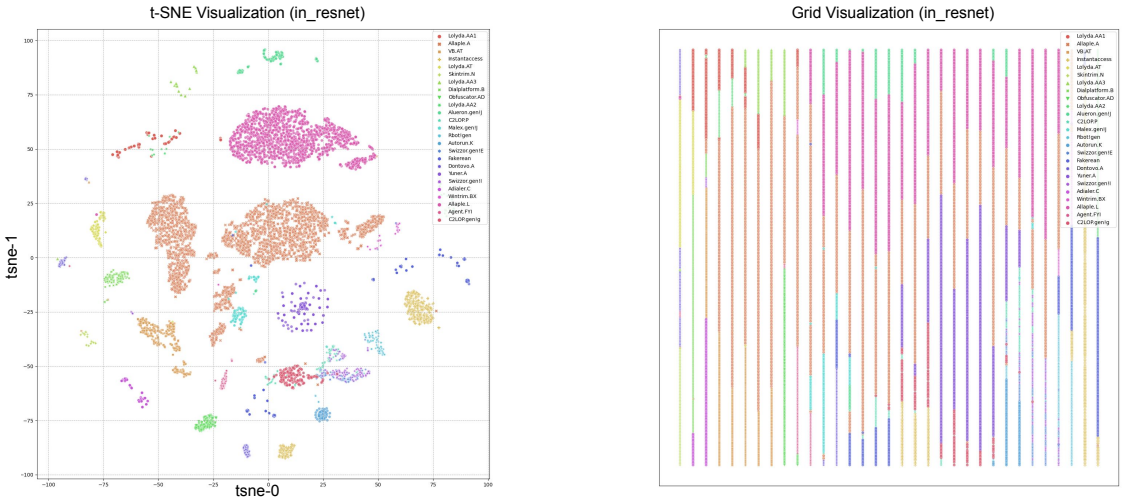
hand-crafted GIST and different deep learning-based features



(a) 2D t-SNE based visualization

(b) 2D Grid-based visualization

Fig. 5: Point-based t-SNE and Grid visualizations of malware (RED) and benign (GREEN) samples from *Smalldata* [31] dataset – Feature: Pretrained-ImageNet (VGG16), Model: RF.



(a) 2D t-SNE based visualization

(b) 2D Grid-based visualization

Fig. 6: Point-based t-SNE and Grid visualizations of 9,339 malware samples from *Malimg* [2] dataset that has 25 different malware families – Feature: Pretrained-ImageNet (ResNet152), Model: RF.

on the dataset. Overall, the "best" feature and classifier are Pretrained-ImageNet (ResNet152) and Random Forest (RF) respectively with an accuracy of `0.987`, macro-precision of `0.970`, macro-recall of `0.965` and macro-f1 of `0.966`. We present the point-based visualizations corresponding to this feature and model in Fig. 6. From the figure, one can easily find nicely formed clusters corresponding to the different (color-coded) malware types. These visualizations with groupings of samples again not only show the efficacy of our approach to visualize malware samples in a grid-like fashion that contain distinguishable clusters, but also support the performance numbers that we get from the multi-class classification experiments.

IV. Spatial Grid-based Visualization of Malware

We mapped the 2D scatter plots to a fixed grid and visualize the malware data as icons or thumbnails on the 2D grid. The steps involved are as follows:

1) Start with a known dataset (malware and/or benign) with fixed number of families and variants, and represent them as grayscale Byteplot images [2].
2) Compute pretrained-ImageNet visual features on these binaries.
3) Reduce the dimensions of the features to 2D using well-known dimensionality reduction techniques such as t-SNE [21].
4) Map the 2D points to a 2D grid, and overlay the thumbnails/icons of the binaries on the grid.

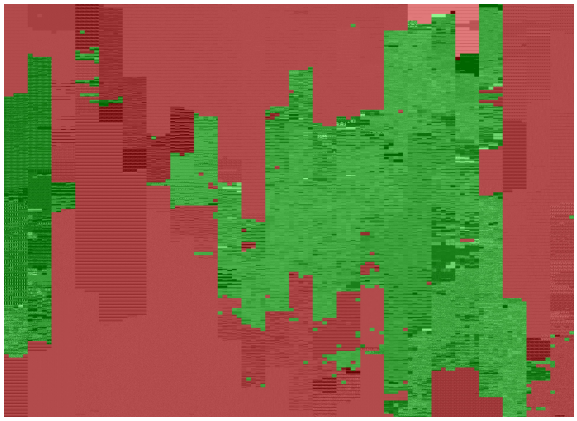


Fig. 7: Visualization of malware (RED) and benign (GREEN) samples from *Smalldata* [31] dataset with Byteplot icons – the malware and benign sets are well separable in feature space.

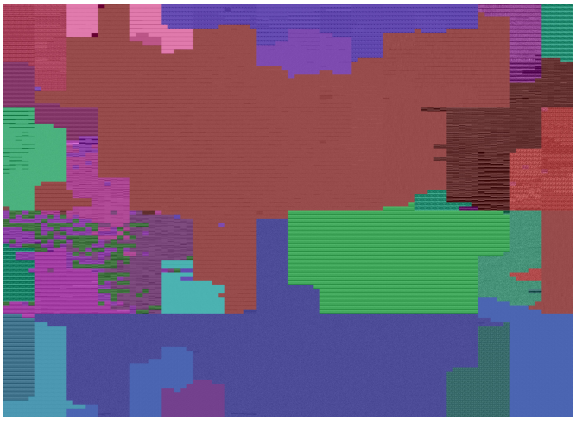


Fig. 8: Visualization of *Malimg* [2] dataset with Byteplot icons – different colored clusters represent well separable malware families in feature space.

A. Visualization of Malware and Benign samples

Here, we visualize malware and benign files from the *Smalldata* [31] dataset on a single grid. Malware thumbnails are shaded in red and benign thumbnails are shaded in green. Fig. 7 shows the grid-based visualization of *Smalldata* dataset. We can clearly see that malware and benign files are clustered well as the red and green regions stand out.

B. Visualization of Malware families

Here, we visualize 25 malware families from the *Malimg* dataset [2] as shown in Fig. 8. As seen in the figure, variants belonging to the same family are clustered close to each other, thus demonstrating the efficacy of grid-based visualization.

C. Case Study: Visualizing the Effect of Packing

We performed a case study to see how grid-based visualizations can help in studying the effect of packed malware variants. For this, we start with a curated malware dataset of unpacked malware variants belonging to 10 different families. Each family has 10 variants, thus totalling 100 samples. Next, we pack these unpacked malware variants using various packers such as *UPX*, *NsPack*, *WinUpack*, *FSG*, *peCompact*, *Polyene*, *Telock* and *Themida*, and thus obtain packed malware variants for different packers. Then, we visualize these packed variants in grids as shown in Fig. 9a, 9b, 9c, 9d, 9e, 9f, 9g and 9h for the malware packers *UPX*, *NsPack*, *WinUpack*, *FSG*, *peCompact*, *Polyene*, *Telock* and *Themida* respectively.

We can see from all these figures that the variants cluster well for packed malware too. The variants appear clear for simple packers like *UPX* while the compression and randomness increases for other packers like *NsPack*, *WinUpack* and *FSG*. Even for advanced compression packers such as *PeLock*, *Polyene* and *Themida*, we can see that the variants clearly form noticeable clusters in the grid. Only for *Themida* which uses compression and encryption, the variants are not as clear, but the clusters are still visible.

V. Conclusion and Future Work

In this paper, we presented two methods to spatially visualize malware samples based on points and icons. The effectiveness of the visualizations are well established from both the malware detection and classification point-of-view. The feature representations used are simple and easy-to-compute that makes the whole visualization process computationally undemanding. Effectiveness of the visualization methods under different malware packing routines further demonstrates its efficacy.

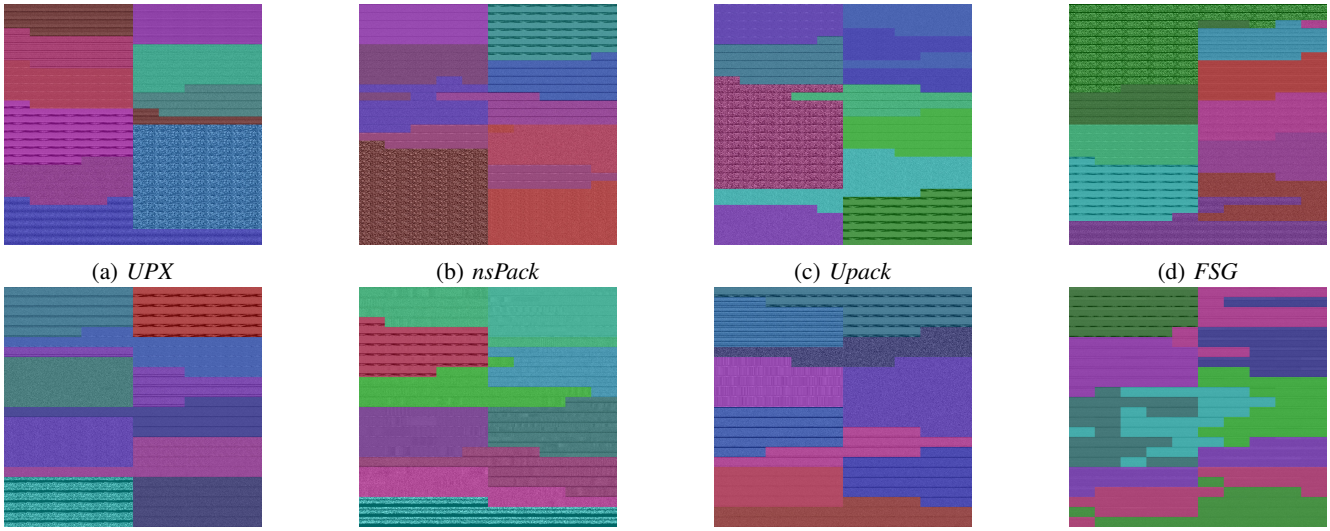
We have restricted our malware dataset platform space to that of Windows. But there are malware datasets that belongs to other platforms including Android and Linux that were not included as part of this comprehensive study. The extended task that includes malware datasets belonging to such other platforms can be part of the future work. We anticipate having similar visual clusters for the malware families. Other deep learning-based features and ensemble models can also be explored in further research. Adversarial analysis that includes specific actions on how a malware author could force the visualization-based misclassification can also be studied. Furthermore, considering 3-dimensional and higher dimensional feature representations can aid in visualizing large scale malware data in the range of hundreds of thousands to millions.

References

[1] "New malicious files discovered daily grow by 5.7% to 380,000 in 2021," https://www.kaspersky.com/about/press-releases/2021_new-malicious-files-discovered-daily-grow-by-57-to-380000-in-2021, 2021. [Online]. Available: https://www.kaspersky.com/about/press-releases/2021_new-malicious-files-discovered-daily-grow-by-57-to-380000-in-2021

[2] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, ser. VizSec '11. New York, NY, USA: ACM, 2011, pp. 4:1–4:7.

[3] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.



(a) *UPX* (b) *nsPack* (c) *Upack* (d) *FSG*

(e) *PeCompact* (f) *Polyene* (g) *TeLock* (h) *Themida*

Fig. 9: Effect of packing on a custom curated malware dataset of 10 malware families – in general, the malware family clusters are still visible in the feature space for the packed variants.

[4] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner, "A survey of visualization systems for malware analysis," in *Eurographics conference on visualization (EuroVis)*, 2015, pp. 105–125.

[5] I. Yoo, "Visualizing windows executable viruses using self-organizing maps," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, 2004, pp. 82–89.

[6] D. A. Quist and L. M. Liebrock, "Visualizing compiled executables for malware analysis," in *2009 6th International Workshop on Visualization for Cyber Security*. IEEE, 2009, pp. 27–32.

[7] P. Trinius, T. Holz, J. Gobel, and F. Freiling, "Visual analysis of malware behavior using treemaps and thread graphs," in *Proceedings of VizSec*, 2009, pp. 33–38.

[8] J. Goodall, H. Randwan, and L. Halseth, "Visual analysis of code security," in *Proceedings of VizSec*, 2010.

[9] Y. Ye, T. Li, Y. Chen, and Q. Jiang, "Automatic malware categorization using cluster ensemble," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 95–104.

[10] A. R. Grégio and R. D. Santos, "Visualization techniques for malware behavior analysis," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense X*, vol. 8019. SPIE, 2011, pp. 9–17.

[11] J. Saxe, D. Mentis, and C. Greamo, "Visualization of shared system call sequence relationships in large malware corpora," in *Proceedings of the ninth international symposium on visualization for cyber security*, 2012, pp. 33–40.

[12] Y. Wu and R. H. Yap, "Experiments with malware visualization," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2012, pp. 123–133.

[13] A. Long, J. Saxe, and R. Gove, "Detecting malware samples with similar image sets," in *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, 2014, pp. 88–95.

[14] H. Kim, J. Kim, Y. Kim, I. Kim, K. J. Kim, and H. Kim, "Improvement of malware detection and classification using api call sequence alignment and visualization," *Cluster Computing*, vol. 22, no. 1, pp. 921–929, 2019.

[15] L. Nataraj and B. Manjunath, "Spam: Signal processing to analyze malware [applications corner]," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 105–117, 2016.

[16] T. M. Mohammed, L. Nataraj, S. Chikkagoudar, S. Chandrasekaran, and B. Manjunath, "Hapssa: Holistic approach to pdf malware detection using signal and statistical analysis," in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 2021, pp. 709–714.

[17] L. Nataraj, T. M. Mohammed, T. Nanjundaswamy, S. Chikkagoudar, S. Chandrasekaran, and B. Manjunath, "Omd: Orthogonal malware detection using audio, image, and static features," in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 2021, pp. 703–708.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[19] N. Bhodia, P. Prajapati, F. Di Troia, and M. Stamp, "Transfer learning for image-based malware classification," *arXiv preprint arXiv:1903.11551*, 2019.

[20] D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, "Image-based malware classification using ensemble of cnn architectures (imcec)," *Computers & Security*, vol. 92, p. 101748, 2020.

[21] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[22] VirusShare, "Virusshare. com–because sharing is caring," 2019.

[23] V. Total, "Virustotal-free online virus, malware and url scanner," *Online: https://www. virustotal. com/en*, 2012.

[24] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[25] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[26] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[27] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 245–250.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] T. M. Mohammed, L. Nataraj, S. Chikkagoudar, S. Chandrasekaran, and B. Manjunath, "Malware detection using frequency domain-based image visualization and deep learning," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021, p. 7132.