Visual Classification of Malware by Few-shot Learning

Kien Tran, Masao Kubo, Hiroshi Sato
Department of Computer Science, National Defense Academy of Japan, 1-10-20 Hashirimizu,
Yokosuka, Kanagawa 239-8686, Japan
E-mail: ed17005@nda.ac.jp, masaok@nda.ac.jp, hsato@nda.ac.jp
www.mod.gov.jp/nda

Abstract

The extent of damage by malware has been multiplying. Many techniques are proposed for detecting malware. However, the usual pattern matching method does not work because when the new malware appeared, many variants are created very soon. In order to catch the new malware, we have to detect and classify them from very few samples. In this paper, we propose a machine learning mechanism that can learn from very few samples of the image of the malware.

Keywords: Few shot Learning, Malware Classification, Matching Network, and Visualization Classification.

1. Introduction

According to a report from Kaspersky Lab, in 2017, at least 360,000 new malicious files were detected every day in 2017 — an 11.5% increase from the previous year[1]. For example, in May of 2017, a new type of ransomware called WannaCry and its variance spread quickly through a number of computers of companies around the world, encrypted files on the PCs and caused substantial financial damage to these companies.

In the report published by Symantec Corporation in February 2019, a decrease in ransomware activity during 2018 had been observed the first time since 2013, with the overall number of ransomware infections on client sides dropping by 20%. However, ransomware such as WannaCry continued to inflate infection figures, and the number of ransomware infections has been shifted toward enterprises with 81% of all infections[2]. These harmful programs are more devastating than others due to their spreading speeds and its functionalities. This trend leads to a need for methods, which are strong enough to detect and classify these kinds of malware as soon as they start spreading widely.

Nowadays, the automatic defense systems can respond to the malware threats by keeping up with the speed of the malware development, but true success depends on strategic insight as well as the speed of the response. Therefore, the most effective defense requires both intelligent (machine learning-led) programs as well as human expertise. Thanks to the huge development of AI technology, more and more fast and reliable methods have been being developed to detect quickly as well as classify new types of malware.

However, there is not always enough samples for the system to learn to recognize new types of malware as the number of new malware has been increasing every day. This is also one of the most key challenges of machine learning solutions is the number of collected samples. Normally, in machine learning, or deep learning, in particular, the more data we collect, the better the accuracy we get.

In this case, an idea of learning object class from only a few data called one-shot/few-shot learning is widely used. Many one-shot learning algorithms have been proposed to deal with the problems of "data-hungry". Therefore, we adopt few-shot learning approach and try to classify malwares from very few samples.

Some Meta-Learning based concepts are introduced in this paper such as a variation of Neural Turing Machine for one-shot learning tasks introduced by A. Santoro et al.[3], Matching Network by O.Vinyals et al.[4], or Siamese Network by G. Koch et al.[5]. These meta-learning models are capable of well adapting or generalizing to new tasks with unknown data using their learned meta-knowledge during training time.

Generally, most of the few-shot learning algorithms have been shown efficiency in the computer vision area. In this paper, a new approach to malware classification with few-shot learning algorithms is introduced. This approach takes advantage of static analysis in which each malware binary code is treated as a greyscale image, then using some state-of-the-art few-shot learning algorithms to classify them.

2. Related works

There are several method to classify malwares by using image of executable binary image. We introduced some examples here. However, these methods need many samples to train their classifiers. This could lead to data hungry problem.

2.1. *Malware Images: Visualization and Automatic Classification*

Using image processing techniques, L. Nataraj et al.[6] proposed an effective method to classify malware. He represented a malware executable as a binary string of zeros and ones. This vector then is shaped into a matrix and viewed as an image. The results showed significant visual similarities in image texture for each malware belonging to the same class. Compare to other traditional classification methods, this approach does not require either malware disassembly or executions, but still showed significant improvement performance. With these images, the authors use GIST to project them into lower dimensions, k-nearest neighbors with Euclidean distance for classification. Since then, some other researches are also introduced. They followed this idea too and used CNN to deal with the classification tasks. Our approaches are also inspired by the work of this image-based malware analysis.

2.2. *Deep learning at the shallow end: Malware classification for non-domain experts*

This research is another approach that considers a malware file as a gray image. This is similar to the image representation of a raw binary file as the work of L.Nataraji et al.[6], but it is simpler. His conversion method preserves the sequential order of the by code in the binaries. He then applies the Convolutional Neural Network in combination with Bi Long Short Term Memory architect (CNN - Bi-LSTM). His approach is applied to the Microsoft Malware Classification Challenge dataset and achieves very good results.

3. Image-based Unknown Malware Classification with Few-Shot Learning Models

In this section, we introduce a meta-learning model to solve the problems of classifying malware classes with very few known samples.

In this approach, since the purpose is to deal with malware classification problems, real behaviors of malware are not necessary to understand. Hence, via static analysis, the contents of the malware file are quickly scanned and visualized as plain pictures. Then, the few-shot models are adapted to classify them in the malware classification problems. The proposed approach is summarized as illustrating in Fig. 1.

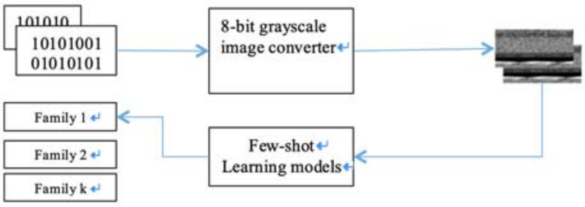


Fig. 1. The proposed approach uses malware binary as an 8-bit grayscale image as input features of few-shot learning tasks.

For Few-shot learning models in Fig.1, we adopt the following two models – Matching Network with external Memory and Weighted Prototypical Class. Both are explained below.

3.1. *Matching Network with external Memory*

To address a challenge of K-shot N-way classification tasks, the proposed models apply embedding learning methods that embed $x \in X \subseteq Rd$ to a smaller embedding space $z \in Z \subseteq Rm$. Using these new spaces, it is easy to identify similar and dissimilar pairs of support samples and test samples. Currently, these methods have three main functions: function *f*(.) embeds sample *xtest* ∈ *Dtest* to *Z*, function *g*(.) embeds *xsupport* ∈ *Dsupport* to *Z* and a similarity measure *s*(.,.) calculates the similarities between the output of f(.) and each output of *g*(.) in the new space *Z*. An overview of this architect is illustrated in Fig. 1.

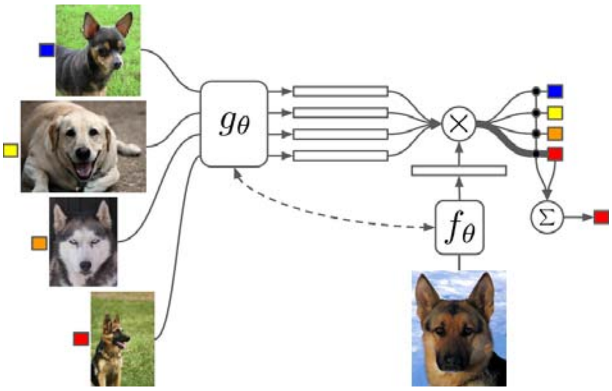


Fig. 2. Illustration of embedding learning methods for few-shot learning classification tasks (excerpt from Ref. 4).

This proposed model draws inspiration from the architecture of the Matching Network model as well as the MANN model for one-shot learning tasks. Thus, its strategy is to enhance an embedding space with memory components, help accordingly recognizing unseen objects based on the content located in these memory matrices.

3.2. *Weighted Prototypical class*

For the N-way K-shot classification tasks, Snell et al.[7] proposed a prototype that computes a representation ck of class k (k=1..N) based on an average calculation of instances of that class.

$$c_k = \frac{1}{K}\sum_{i=1}^{K} g(x_{k,i}) \tag{1}$$

In some cases, the class distribution is skewed. That is, some samples could locate outside the range of major samples in the class. Appling a prototype from eq (1) in such a situation could lead to a biased mean sample of the class. One way to overcome this is to treat those samples unequally based on their weights. These weights, which are used to determine the relative importance of each data point, are considered as the distance of a point to other points in the same class. So, the contributions of the points to the representative point of their class are proportional to the distance between them and the others.

4. Experiments

In this chapter, we will perform 5-way 1-shot tasks and 5-way 4-shot tasks using the approach with two few-shot models (i.e., Matching Networks and Prototypical Networks) on the datasets called MalImg.

We based on meta-learning methods to classify malware. First, we assume some malware classes in MalImg are already known. These samples are trained with few-shot learning models. Then, they are tested with the rest classes which are assumed as never-seen-before classes. We also use the results extracts from the experiment of Kang et al.[8] as the baselines for our comparison.

4.1. *Dataset*

To demonstrate his approach, Nataraj et al.[6] introduced a large dataset of 25 families with more than 9400 malware. The provided samples are stored as greyscale images with different dimensions according to their original file size. The detail of this dataset is summarized in Table 1.

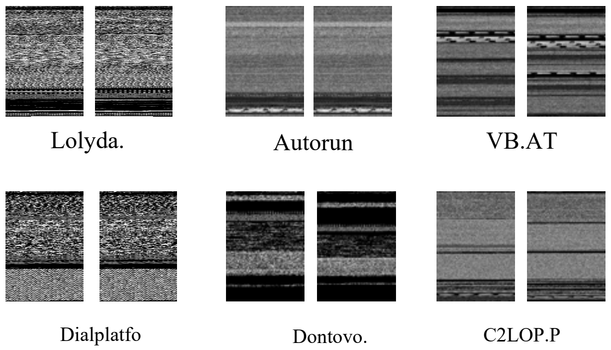
Table 1. MalImg dataset Description

Family	Samples	Family	Samples
Allaple.L	1591	Alueron.gen!J	198
Allaple.A	2949	Malex.gen!J	136
Yuner.A	800	Lolyda.AT	159
Lolyda.AA1	213	Adialer.C	122
Lolyda.AA2	184	Wintrim.BX	97
Lolyda.AA3	123	Dialplatform.B	177
C2Lop.P	146	Dontovo.A	162
C2Lop.gen!g	200	Obfuscator.AD	142
Instantaccess	431	Agent.FY!	116
Swizzot.gen!!	132	Autorun.K	106
Swizzor.gen!E	128	Rbot!gen	158
VB.AT	408	Skintrim.N	80
Fakerean	381		

Regarding malware families, malware authors usually develop new malware based on their previous codes. Only small parts of the old malware are rewritten or removed. By visualizing all parts of the binary file as an image, the analysts themselves by empirical observation could easily recognize the differences between the malware belonging to different classes.

To visualize a malware as an image, we transform a given malware binary file as vectors of 8-bits unsigned integer. Each vector represents a pixel value of the target greyscale image, which is in a range of 0 and 255. Finally, based on the malware file size indicated in the works of Nataraj et al.[6], the resolution of the image is decided. Hence, if a small portion of code is changed, the overall patterns of the malware families may not be affected.

Fig. 3 illustrates the images of specific families of malware. As can be seen from Fig. 3, various malware families have distinct visual characteristics.


Lolyda.

Autorun

VB.AT

Dialplatfo

Dontovo.

C2LOP.P

Fig. 3. Some visualization examples extracted from various families. These images from the MalImg dataset indicate that the difference between classes could be easily distinguished by observation.

4.2. *Setting*

In these experiments, since we want to simulate the situation in which the models could classify malware into certain families with a few knowledge of them, The dataset is split into three parts, one part is used for training the model with 11 families, other 7 families are for the validating, and the rests are for testing model. Among available classes, we randomly select five classes then perform 5-way classification tasks.

Two experiments are implemented. The first scenario is only one sample per class provided. The models have to guess the class of the test sample. This scenario is called a 5-way 1-shot task. Another experiment is a 5-way 4-shot task in which instead of using one sample per class, four random samples are picked out of five random classes. For every malware inputs belonging to those five classes, the models have to guess their family. These experiments are implemented with both Matching Networks and Prototypical Networks.

To begin the experiments, we resize all samples into an 84x84 scale despite their varied sizes. These samples are extracted the necessary information and embedded into feature space via a simple yet powerful Convolution Neural Network (CNN). This CNN consists of four stacked blocks of {3×3-convolutional layer with 64 filters, batch-normalization, 2×2 max-pooling, leaky-relu, drop out layer with rate 0.3}. The output is passed through a fully connected layer resulting in a 64-dimensional embedding output.

The 64-dimensional vectors are then fed to function $f(.)$ and function $g(.)$ in the Matching Networks (MNets) model to compute the Cosine distance between the support samples and test sample.

In the case of the Prototypical Networks (ProtoNets), the prototype of each supported class is computed as an average of all 64-dimensional vectors of that support class. Those calculated prototypes are used to classify the test samples based on the L2 Euclidean distance between them and the test samples.

4.3. *Result*

The experiment results reported in Table 2 are the averages of 1,000 times of testing. The baselines are extracted from the works of Kang et al.[8].

To classify malware to certain families based on one or a few known samples, we use the visual similarity of malware images with some few-shot models, which have been well studied. The two few-shot learning models used in this approach are Matching Networks[4] and its extended variation, Prototypical Networks[7].

It is seen that Matching Networks and Prototypical Networks overcome the model using Memory Augmented Neural Network and the other baselines in both 1-shot and 4-shot tasks. Moreover, the differences between these results are very high. While the MANN could perform only 66.2% accuracy on 1-shot task and 79.4% on 4-shot task, both the MNets and the ProtoNets are all reaching over 86% and 89% respectively. Especially, with the ProtoNets, even only one sample is provided, the model could determine the correct class among five classes of test samples with 92.4% accuracy. If four samples of each class are known, the model is superb with 95.3% certainty.

Table 2. MalImg dataset's Classification Results

Model	1 shot	4 shots
FeedForward Network	37.9%	30.0%
CNN	42.6%	50.6%
LSTM	56.2%	64.1%
MANN	66.2%	79.4%
Matching Networks	86.3%	89.7%
Prototypical network	**92.4%**	**95.3%**

5. Conclusion

This paper proposed another malware classification approach that takes advantage of the developments of few-shot learning algorithms to introduce a novel way of malware classification. It helps malware analysts to quickly classify malware into the correct groups even with only a few known samples. In this approach, the effectiveness of classifying malware based on Image Processing in combination with other few-shot learning models (Matching Networks and Prototypical Networks) has been proven. The accuracies of classifications are outstanding, even with only one provided sample. Furthermore, the accuracies of the classifications could be improved by adjusting the hyper-parameters of the embedding network (CNN) as well as Image Processing procedures.

As future work, it is necessary to dig deeper into this method to reduce the effect of noise samples, and improve final results as well as compare them to existing methods in case of few-shot learning tasks. We will also take a more in-depth look at some other one-shot learning algorithms or other simpler methods such as NCC (Normalized Cross-Correlation) to find more suitable methods for malware analysis to improve the accuracies of our approach. More benign programs will also be collected along with different kinds of malware; hence, we could re-evaluate better our methods.

References

1. Kaspersky, "Kaspersky Lab Report", https://www.kaspersky.com/about/press-release/2017kaspersky-lab-detects-360000-new-malicious-files-daily, accessed Nov 10, 2019.
2. Department of Health and Care, Securing cyber resilience in health and care Progress update October 2018, Oct 2018, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/747464/securing-cyber-resilience-in-health-and-care-september-2018-update.pdf, accessed Dec 12, 2019.
3. A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. "Meta-learning with Memory Augmented Neural Networks." In Proceedings of The 33rd International Conference on Machine Learning, pp. 1842–1850, vol. 48, 2016. R. Loren and D. B. Benson, Deterministic flow-chart interpretations, *J. Comput. System Sci.* **27**(2), 1983, pp. 400–433.
4. O. Vinyals, C. Blundell, T. Lillicrap, k. Kavukcuoglu, and D. Wierstra. "Matching Networks for one-shot learning," In Advances in Neural Information Processing Systems, pp. 3630– 3638, vol. 29, 2016.
5. G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," In ICML 2015 Deep Learning Workshop, 2015.
6. Zhenguo Li and Fengwei Zhou and Fei Chen and Huang Li, "Meta-SGD: Learning to Learn Quickly for Few-Shot Learning", ArXiv, abs/1707.09835, 2017
7. J. Snell, K. Swersky, and R. Zemel."Prototypical networks for few-shot learning," In Advances in Neural Information Processing Systems, pp. 4077–4087, vol. 30, 2017.
8. Min Chul Kang, Huy Kang Kim, "Rare Malware Classification Using Memory Augmented Neural Networks," (Korean) Journal of The Korea Institute of Information Security & Cryptology, vol. 28, No.4. pp. 847–857, Aug 2018.