TECHNICAL SESSION 1 (13:00~14:15)

• BLE-based Localization of Smartphone for the Passive Entry and Passive Start System
/ Kyung-Hyun Koo, Lynn Choi
/ Korea University / Korea

• A Creative Computing Approach to Scaling Knowledge based on A 4D-Knowledge Coordinate System
/ Lin Zou, Hongji Yang
/ Bath Spa University / England

• Usability of Docker-based Container System Health Monitoring by Memory Dump Visualization
/ Sang-Hoon Choi, Daeseon Choin, Ki-Woong Park
/ Sejong University, Kongju National University / Korea

• Improvement of NTRUEncrypt using Sliding Window Methods
/ Won-Il Pyo, Mun-Kyu Lee
/ Inha University / Korea

• A language assistant system for smart glasses
/ Shi Fan Zhang, Kin Hong Wong
/ The Chinese University of Hong Kong / Hong Kong

Usability of Docker-based Container System Health Monitoring by Memory Dump Visualization

Sang-Hoon Choi
SysCore Lab.
Sejong University
Seoul, Korea
csh0052@gmail.com

Daeseon Choi*
Dept. of Medical Information
Kongju National University
Chungnam, Korea
sunchoi@kongju.ac.kr

Ki-Woong Park*
Dept. of Computer and Information Security
Sejong University
Seoul, Korea
woongbak@sejong.ac.kr

***Abstract*— Recently, works to apply machine learning technology to system monitoring field are actively being carried out. In this direction, we have conducted a work to classify the state of the system using memory visualization and deep learning technology. In this work, we visualize the memory data of the operating system on container engine and learning using CNN algorithm which is specialized to classify the image. As a result of our experiments, it is possible to completely classify the state of the memory after two learning processes for 1,000 training data. Furthermore, when a 10% blur effect was applied to the visualized image file, our classification model can classify the state of the memory with 81.1%. Through these experiments, we verified that learning technology using binary visualization can be applied to system health monitoring field.**

Keywords—System Monitoring, Data Visualization, Deep Learning

I. Introduction

Recently, many works have been conducted due to the improvement of algorithm performance of artificial intelligence technology [1, 2]. There is also the work to integrate artificial intelligence into system monitoring field [3, 4]. As typical examples, there are the studies on a method of detecting a spoofing attack and analyzing the operation of the process through a packet on a network [5, 6]. However, such a machine learning-based monitoring studies have a limitation that requires data pre-processing.

In this paper, we try to utilize the operating system memory for system health monitoring and learn the memory data using CNN (Convolutional Neural Network) algorithm [7]. However, there is a problem that it is difficult to collect and learn the operating system memory because it has a relatively large size compared to other learning data. In order to solve these limitations, we have extracted the memory data through our previously studied operating system memory high speed dump technology and visualized the extracted memory to minimize the size of data required for learning [8].

The overall framework for visualizing and learning the memory in this work is shown in Fig 1. In order to verify that the visualized memory data can be utilized as a system monitoring element, we visualized the memory data on web service which and the memory data on web service which is abnormally running and learned it to CNN algorithm. As a result of our experiments, it is possible to completely classify the state of the memory after two learning processes for 1,000 training data. In addition, a blur effect was applied to the image to measure the classification performance of the memory image when modulation of the image occurred. When a 5% blur effect was applied to an image file classification model can completely classify the state of the memory. However, when the blur effect of the image file was 10%, our model dropped the classification accuracy of the memory state to 81.1%.

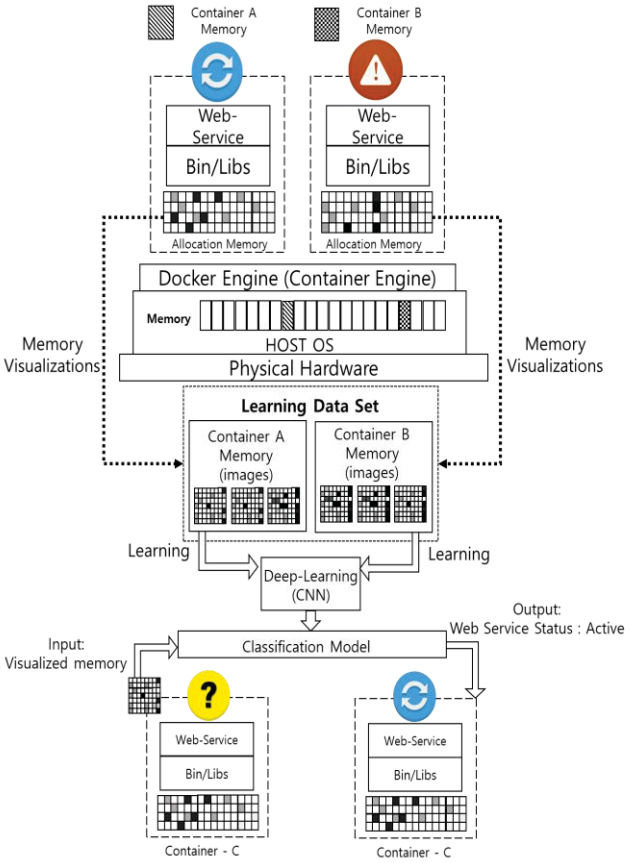

Fig. 1. Memory Learning Framework for System Status Extraction

*: Corresponding Author (Ki-Woong Park, woongbak@sejong.ac.kr)
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2017R1C1B2003957, NRF-2016R1A4A1011761)

Composition of this paper is as follows. Section 2 explains the binary learning work using machine learning. Section 3 explains the learning framework through memory visualization. Section 4 explains the experimental results on the learning performance of the memory visualization technique. The last section explains the conclusion.

II. Related Work

Works have been steadily conducted to visualize binary data as significant results. In particular, works on malicious code analysis using binary data of malicious codes are actively progressing. In 2011, L. Nataraj proposed a work to visualize and classify malicious code files [9]. He classification scheme proposed by L. Nataraj showed about 4,000 times faster performance than the existing malicious code dynamic analysis technology.

In 2015, Microsoft hosted a conference on technology to improve the accuracy of 9 Family Classifications by disclosing data from its vaccine server on Kaggle [10] and analyzing public malware samples. As a result of the conference, studies that visualized only binary data of malicious code showed high accuracy for family classification. Mansour Ahmadi proposed a method for the most accurate family classification [11]. The work showed that nine malicious codes were correctly classified into the same type with 99.8% by using learning method of visualizing the binary of the malicious codes and learning method of hybrid symbol, metadata and entropy. The works to obtain significant results through deep learning by imaging binary data can also be applied in the field of system monitoring in the future.

III. Memory visualization and learning framework for usability verification

This chapter explains the advantages of memory visualization technology and suggests how to visualize and learn memory.

A. Learning through binary data visualization

Work has been actively conducted to apply deep learning to monitoring technology. For example, it is typical to image and classify binary or assembly language of malicious code. As another example of this, security technology work through memory monitoring is actively being carried out [11]. Monitoring the memory area can detect malicious code or respond to malicious code such as Ransomware [12]. In particular, in environments where resources are virtualized, such as in a cloud computing environment, memory analysis technology is highly utilized. The reason for the high utilization of this technology is that the system memory contains all the behavior data. A memory dump is necessary to analyze the system memory in detail.

However, in the case of a memory dump file, the size of the system's physical memory is the same as the size of the dumped file, so it takes a large amount of disk space to free the dumped file and long time to analyze the memory. In order to solve these limitations, we have extracted the memories through our previously studied operating system memory high speed dump technology and visualized the extracted memory to minimize the size of data required for learning. Visualization of memory data makes it easier to manage and analyze data because it has the advantage of significantly reducing memory size. Thus, 3bytes of memory can be converted into one pixel.

B. Memory Visualization Scheme

In this paper, we apply a method to image data of memory by using Red, Green, and Blue colors, called 3 channels. We read 3bytes (24bit) of memory binary to represent 3channel as one pixel. To express the read value in pixels, we change the binary of 1 to 8 bits to Integer; calculate the integer is mod 256, set the value derived from mod 256 to Red. Equally, we read the binary of 9 to 16 bits, change the binary value to integer, and calculate the integer is mode 256, set the value derived from mod 256 to Green. We read also the binary of 17 to 24bits and have Blue value through the same process.

IV. Experiments

In this chapter, we measure the classification performance of classification model after converting the memory at active state's web service and the memory at stop state's web service into images respectively, and train it into CNN algorithm. Furthermore, the accuracy of the classification when the learned images have blur effect is measured.

A. Experiment Environments

We conducted the following experiments. We used ubuntu 16.04, 32GB of memory, normal HDD as storage server and used Ubuntu 16.04 64bit as container using Docker [12]. And we installed apache inside the container to run the web service. Finally, we performed a dump of 500 times in a cycle of 5,000ms each for a state in which a web service can be provided and a state in which a web service cannot be provided. The detailed experimental environment is shown in Table I.

TABLE I. Details of Environments

Name	Spec
OS	Ubuntu 16.04 64bit
CPU	Intel® Xeon® CPU E5-2609 v2 @ 2.50GHz
RAM	32GB
Language	Python 3.5
Tensorflow	1.1.0
Keras	2.0.8
Container	Ubuntu 16.04
Web Service	Apache 2.4.7

B. *Classifications model Evaluations*

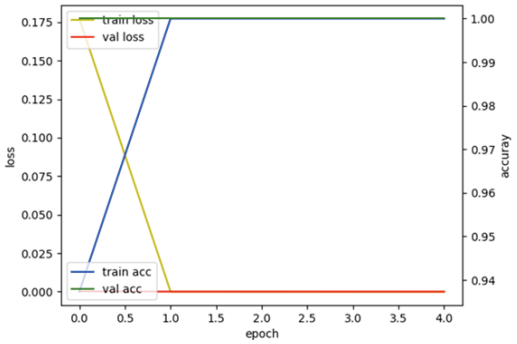

Fig. 2. Learning process of visualized memory data

The classification performance obtained by learning the visualized memory through CNN algorithm is shown in Fig. 2. The *train_loss* (yellow) is the training loss value, x axis is the number of epochs, y axis is the loss value, *ver_loss* is the verification loss *valu*, *train_acc* is training accuracy and *ver_acc*(green) is the verification accuracy. We classification model, which we modeled, was able to accurately classify the active and stop state of the current web service as 100% in only two image learning.

C. *Measurement of classification performance according to images modulation*

TABLE II. Bluer Effect Evaluations

Bluer	Size (bytes)	Compress	Accuracy
0%	433,254	0%	100%
5%	391,3778	90.3%	100%
10%	351,630	81.1%	100%
20%	227,302	64.0%	74%
50%	108,734	25.1%	51%

To verify the accuracy of the model learned, we measured the classification performance with the memory image file modulated. In this experiment, modulation of image file means blur effect. The experimental results are shown in Table II. When a 5% blur effect was applied to an image file with a size of 433.254 bytes, the size of the image file was compressed by 90.3% and our model can completely classify the state of the memory. The size of the image file that has 10% blur effect was compressed to 81%, and classification model was able to completely categorize the state of the memory.

However, when the blur effect of the image file was 20%, our classification model dropped the classification accuracy of the memory state to 75%. Finally when the blur effect of the images file was 50%, classification of the memory based on the learned data was impossible.

V. Conclusion

In this paper, we demonstrate the usability of monitoring system health through visualization of memory data. To verify the usability of the memory visualization technology, we extracted the memory data on web service that is active, converted the data into the image, and training it through CNN algorithm. As a result, the designed scheme was able to distinguish between active memory and stop memory on web service with 100% accuracy. Also, experiments demonstrate that even if the visualized memory is falsified, Stop of active states of web service can be distinguished with a high probability. As our further work, we will conduct work on the most efficient image processing technique for learning memory data according to user's purpose.

References

[1] Andrieu, Christophe, et al. "An introduction to MCMC for machine learning." Machine learning 50.1-2 (2003): 5-43.

[2] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.

[3] Athanasiadis, Ioannis N., and Pericles A. Mitkas. "An agent-based intelligent environmental monitoring system." Management of Environmental Quality: An International Journal 15.3 (2004): 238-249.

[4] Low, Yucheng, et al. "Distributed GraphLab: a framework for machine learning and data mining in the cloud." Proceedings of the VLDB Endowment 5.8 (2012): 716-727.

[5] Rieck, Konrad, et al. "Automatic analysis of malware behavior using machine learning." Journal of Computer Security 19.4 (2011): 639-668.

[6] Narudin, Fairuz Amalina, et al. "Evaluation of machine learning classifiers for mobile malware detection." Soft Computing 20.1 (2016): 343-357.

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[8] Choi, Sang-Hoon, Yu-Seong Kim, and Ki-Woong Park. "Toward Semantic Gap-less Memory Dump for Malware Analysis." ICNGC Conference, Jenuary. 2016.

[9] Nataraj, Lakshmanan, et al. "Malware images: visualization and automatic classification." Proceedings of the 8th international symposium on visualization for cyber security. ACM, 2011.

[10] Kaggle: https://www.kaggle.com/

[11] Ahmadi, Mansour, et al. "Novel feature extraction, selection and fusion for effective malware family classification." Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. ACM, 2016.

[12] Merkel, Dirk. "Docker: lightweight linux containers for consistent development and deployment." Linux Journal 2014.239 (2014): 2.

.