International Journal of Engineering & Technology, 7 (3.7) (2018) 511-513

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Malware Classification by Ensemble Application of Convolutional and Recurrent Neural Networks

Hae-Jung Kim[1*]

[1]*Department of Cyber Security, Kyungil University*
50, Gamasil-Gil, Haynang-Eup, Gyeongsan-Si, Gyeongbuk, 38428 Korea
**Corresponding author E-mail: E-mail: hjkim325@kiu.kr*

Abstract

Malicious software written for malevolent purposes poses a serious threat to information security. With respect to information security for malware treatment, malicious codes must be correctly classified. In this paper, we propose an ensemble classification scheme for the convolutional neural network and recurrent neural network models. We then analyze the classification results of malicious software. These results are presented as a confusion matrix and cosine similarity. The performances of the classifiers are compared and visualized by using graphical representations. The performance of the proposed ensemble model was the highest at 96.50%, indicating its viability as an accurate classification model.

Keywords: *Malware classification; convolutional neural network (CNN); recurrent neural network (RNN)*

1. Introduction

In the present era of big data, social network services and smartphones are producing an exponential increase of data [1]. Big data includes a huge amount of personal and financial information, which should be absolutely protected through appropriate security measures. Malware written with malicious intent has become a serious threat to information security.

Classifying malware accurately is a priority objective for resolving it and achieving the fundamental security of information. In other words, an adequate protection method can be provided by detecting and analyzing malware, considering the fact that malware classification is an emerging global issue [2]. Moreover, since the volume of big data is well beyond the analytical capacity of the human brain, the importance of artificial intelligence (AI) is increasing. In this paper, the framework of an ensemble AI technique combining a convolutional neural network (CNN) and a recurrent neural network (RNN) is proposed to learn and classify malware quickly and accurately. A malware data set is used to compare and verify the performance of different classifiers.

2. Literature Review

Malicious software (malware) is an umbrella term referring to each program and its components that are written with malicious intent. The signature-based method, which is most widely used for malware analysis, analyzes and identified features of existing malware [3]. However, since only a slightly different input value can escape hashing values, this method may not detect new malware. Another method involves executing malware directly in cyberspace to analyze behavior patterns. A combination of these two methods also exists [4]. Consequently, new classification techniques that are more flexible and accurate are needed for accurate analysis.

Various classification models using logistic regression, decision trees, RNNs, etc., have been proposed for malware classification [5]. Many researchers, including Kolter and Maloof, compared malware classifications using a Bayesian network, decision trees, and support vector machines [6]. Along with malware image research, some studies have addressed variants of Echo state network (ESN) and RNN [7] [8]. If a malware detection system using a machine-learning-based classifier employs too many features, it is vulnerable to unnecessary information that is used for the input of classifiers, thereby degrading the classification performance [9] [10]. This research strives to improve the classification accuracy by using an ensemble classification method that applies CNN, which shows the best performance for image processing, to learning. It reduces the dimension during preprocessing for feature extraction, when imaging by pattern recognition. Moreover, it applies the long short-term memory (LSTM) algorithm from among RNNs for classification.

3. System Architecture and Ensemble Technique

3.1 System Architecture

The proposed deep-learning-based ensemble classifier consists of two parts. The first part uses the convolution-pooling operation for modeling a preprocessed malware image; the second one is used for modeling the sequence of malware by using an LSTM neural network, which is the most widely known among RNNs for complex sequence modeling.

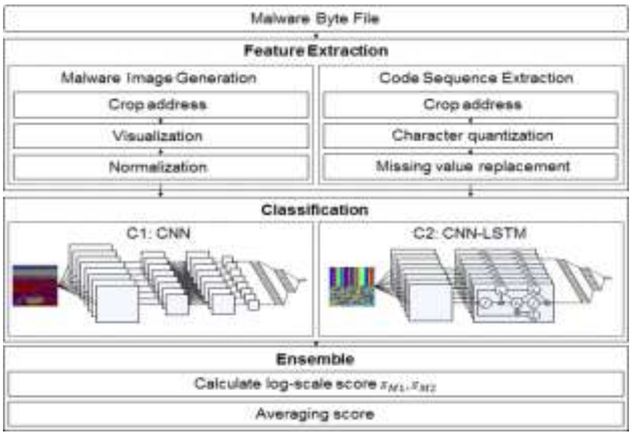
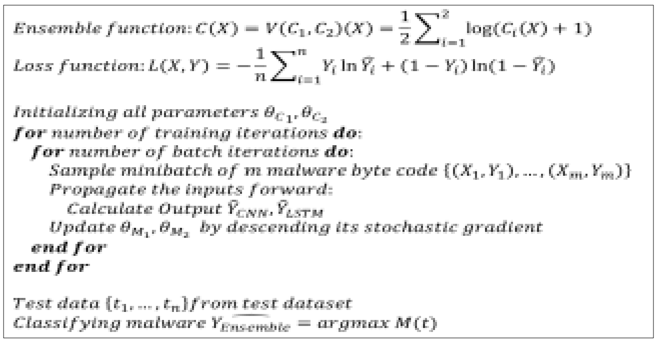


Figure 1: Architecture of the proposed model with an ensemble of CNN and LSTM

Figure 1 shows a schematic diagram of the architecture of the entire model and the ensemble technique. The proposed classifiers are named C1 and C2, respectively. These classifiers have a clear advantage in the malware classification domain, and were thus designed to reflect the complementary relation. Each classifier outputs a sigmoid activation function value, as shown in (1), by multiplying a vector, which is the output at the $(l-1)$th layer, and p^{l-1} by the weight of the i-th node w_i^{l-1}.

$$\hat{y}^l = w_i^{l-1}(\sigma(p_i^{l-1}) + b_i) \tag{1}$$



```
$Ensemble\ function: C(X) = V(C_1, C_2)(X) = \frac{1}{2}\sum_{i=1}^{2} \log(C_i(X) + 1)$
$Loss\ function: L(X, Y) = -\frac{1}{n}\sum_{i=1}^{n} Y_i \ln \hat{Y}_i + (1 - Y_i)\ln(1 - \hat{Y}_i)$

Initializing all parameters $\theta_{C_1}, \theta_{C_2}$
for number of training iterations do:
  for number of batch iterations do:
    Sample minibatch of m malware byte code $\{(X_1, Y_1), \dots, (X_m, Y_m)\}$
    Propagate the inputs forward:
      Calculate Output $\hat{Y}_{CNN}, \hat{Y}_{LSTM}$
    Update $\theta_{M_1}, \theta_{M_2}$ by descending its stochastic gradient
  end for
end for

Test data $\{t_1, \dots, t_n\}$ from test dataset
Classifying malware $Y_{\widehat{Ensemble}} = argmax\ M(t)$
```

Figure 2: Learning and althrothm of the ensemble model

Figure 2 presents the learning and algorithm of the proposed ensemble model. The loss function L is defined by the difference of the probability distribution between the malware class vector $\hat{Y}$, which is the output from each classifier, and the real malware class vector Y.

The output vectors C_1(X) and C_2(X) quantify the probability that an input image will belong to each malware type, by a number between zero and one. To synthesize each classifier from a complementary perspective, the output of each model is emphasized by a log scale, as given in (2); the arithmetic mean is calculated, and vector C(X) of the ensemble model is outputted.

$$C(X) = V(C_1, C_2)(X) = \frac{1}{2}\sum_{i=1}^{2} \log_2(C_i(X) + 1) \tag{2}$$

The classification process begins with the conversion of the elements of each output vector $C_i(X)$ of each model, which is outputted by a probability between zero and one into a log scale. It is completed by selecting the largest element of $C(X)$ that is averaged from the outputs of the two models. The confusion matrix analysis is used for verification.

3.2 Generalization Performance using the Convolution pooling Operation

Convolution and pooling, which are applied to the proposed CNN to classify malware, distort the input images and decrease resolution, thereby producing a more robust generalization performance compared to that of other AI algorithms.

The element $C_{xy}^l$of the vector C^l, which is the output at the l-th convolution layer, conducts a convolution operation of (3) using the output vector of the previous layer y^{l-1} and filter w, which is a vector with a magnitude of $m \times m$. The filter, being smaller than the image, is multiplied by the output vector of the previous layer, and thus distorts a part of the malware image.

$$C_{xy}^l = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1} w_{ab}\, y_{(x+a)(x+b)}^{l-1} \tag{3}$$

At the l-th pooling layer, a maximum value-pooling operation of equation (4) is conducted to select a maximum value as a representative value from the $k \times k$ region of the vector with a magnitude of $N \times N$, which had been input at the previous layer, and then a vector with a magnitude of $\frac{N}{k} \times \frac{N}{k}$ is outputted. τ is the pooling distance of the region where the pooling operation is performed.

$$p_{xy}^l = max_{r \in R}\, c_{xy \times \tau}^{l-1} \tag{4}$$

When the malware image is decomposed by repeating the convolution-pooling operation and arrives at the l-th full connection layer at the bottom of the classifier, the value of each node is determined by (5). w_{ji}^{l-1} is the product of the i-th node of the $(l-1)$-th layer and j-th node of the l-th layer. The vector p_i^{l-1}, which is the output of a nonlinear sigmoid function σ at the last pooling layer, is added with a bias b_i and is then multiplied by w_{ji}^{l-1}.

$$h_i^l = \sum_j w_{ji}^{l-1}(\sigma(p_i^{l-1}) + b_i) \tag{5}$$

Then, a learning process is conducted by using an error backpropagation algorithm to modify the synaptic connection (weight) inside a classifier, so that each input image can be mapped onto a malware type that needs to be classified.

3.3 LSTM for Sequence Modeling

The Microsoft malware classification challenge dataset, which was used as the learning and verification data for the proposed artificial intelligence deep learning based malware detection system, was presented at the Kaggle machine learning challenge a machine learning based data analysis contest hosted by Microsoft in 2015 [3]. A total of 10,868 malware comprising nine different types and approximately 200 GB are shown in Table 1.

LSTM is an RNN model that is suitable for different types of sequence data processing. Unlike typical RNNs, LSTM consists of flexible cells that can adjust the degree values of the input, output, and storage.

LSTM is used to repeatedly calculate unit activation and to calculate the mapping from the input to the output sequences. If a typical RNN alone is used for malware classification, high accuracy can be achieved; however, the extraction of one malware feature is a time-consuming task. Thus, an ensemble of two classifiers is used.

4. Experiment and Evaluation

The Microsoft malware classification challenge dataset, which was used as the learning and verification data for the proposed artificial intelligence deep learning based malware detection system, was presented at the Kaggle machine learning challenge a machine learning based data analysis contest hosted by Microsoft in 2015 [3]. A total of 10,868 malware comprising nine different types and approximately 200 GB are shown in Table 1.

Table 1. Malware type and features in the data set

Class index	Malware name	Description
1	RAMIT	RAMIT
2	Good Similar	Very well
3	KELIHOS v.3	p2p botnet using polymorphism Encrypted
4	VUNDO	Multi-component malware family: trojan, worm
5	SIMDA	Most complex malware, Multi-component malware family: botnet,trojan,backdoor, password-stealing
6	TRACUR	Trojan
7	KELIHOS v.1	Botnet
8	OBFUSCATOR.ACY	Combination of methods: Encryption, Compression,Anti-debugging, Anti-emulationtechniques
9	GATAK	Trojan

The malware classification of the proposed model was experimentally evaluated. The classification performance was verified by analyzing the confusion matrix and cosine similarity among the classified malware images.

The Kaggle Microsoft Malware Classification Challenge (BIG 2015) was used as the data set of the malware classification experiment [11]. Of the 10,866 labeled malware data, 70% was used as learning data, 20% was used as verification data, and the remaining 20% was employed as test data.

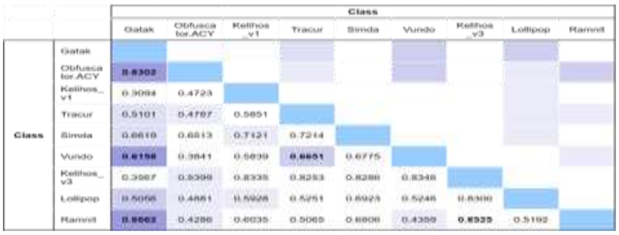
The 400-GB data set provided by BIG 2015 was used to conduct the learning of the most widely known nine types of malware, thereby proposing nine classifications of malware. In addition, the performance was analyzed through a ten-fold cross validation; the analysis results are presented in Table 2.

The malware classification performance of the proposed CNN-LSTM ensemble model was 96.50%, indicating an improvement from the CNN's 95.42% and LSTM's 94.89%.

Table 2: Performance evaluation using ten-fold cross validation

Index	CNN	LSTM	Ensemble
1	0.9634	0.956	0.9653
2	0.9538	0.9528	0.9761
3	0.9588	0.9516	0.9591
4	0.9606	0.9495	0.9799
5	0.9597	0.9505	0.9618
6	0.9565	0.9538	0.9655
7	0.9588	0.9517	0.9734
8	0.9514	0.9546	0.9666
9	0.919	0.9167	0.9387
10	0.9602	0.9519	0.9634
Avg.	0.9542	0.9489	0.9650
Stdev.	0.0129	0.0115	0.0113

As a next step, the cosine similarity between the two malware vectors was used to analyze the misclassified data. Figure 4 shows a colored chart illustrating the analysis results of the misclassified data produced through cosine similarity.



		Class								
		Gatak	Obfuscator.ACY	Kelihos_v1	Tracur	Simda	Vundo	Kelihos_v3	Lollipop	Ramnit
Class	Gatak									
	Obfuscator.ACY	0.8302								
	Kelihos_v1	0.3094	0.4723							
	Tracur	0.5101	0.4707	0.5851						
	Simda	0.6610	0.6813	0.7121	0.7214					
	Vundo	0.8158	0.3841	0.5839	0.8651	0.6775				
	Kelihos_v3	0.3567	0.8396	0.8335	0.8253	0.8288	0.8348			
	Lollipop	0.5056	0.4881	0.5926	0.5251	0.6923	0.5246	0.6300		
	Ramnit	0.8663	0.4286	0.6035	0.5065	0.6808	0.4350	0.8535	0.5192	

Figure 4: Similarity of misclassified data

The average similarity of up to 100 malware images is displayed by quantifying the similarity between two vectors. The similarity is proportional to the number of misclassifications. The similarity between Gatak and Ramnit is the highest, and the number of the corresponding misclassifications was the highest from the experiment.

5. Conclusion

Nine classifications were created based on the nine types of malware, which were the most widely known, as experimental data. The performance of each AI deep-learning model was verified. In addition, the proposed ensemble classifier combining CNN and LSTM was compared with CNN and LSTM. The complementary relation resulted in an improvement in the performance. It is expected that the AI deep-learning-based neural network will be expanded to form an ensemble with a new model and provide enhanced results in the future.

On the other hand, sufficient data set must be secured for application as big data and for learning. It is necessary to study the challenges related to the fact that malware executes multiple malicious functions simultaneously. Various methods of improving the accuracy of malware classification are also needed.

References

[1] Chen, M., et al., *Big Data Analysis. Big Data*. Springer International Publishing, 2014. : p. 51-58. http://www.springer.com/in/book/9783319062440

[2] Luo, X., and Liao, Q. Awareness Education as the key to Ransomware Prevention. *Information Systems Security*, 2007. *16*(4): p. 195-202.

[3] Sathyanarayan V. S., Kohli P., Bruhadeshwar B., Signature Generation and Detection of Malware Families. Information Security and Privacy. Springer, Berlin, Heidelberg. ACISP 2008. 5107. : p. 336-349 https://doi.org/10.1007/978-3-540-70500-0_25

[4] Damodaran, A., et al. A comparison of static, dynamic, and hybrid analysis for malware detection. *Journal of Computer Virology and Hacking Techniques*, 2017. *13*(1): p. 1-12.

[5] Dahl, E., Stokes, J. W., Deng, L., and Yu, D. Large-scale malware classification using random projections and neural networks. Paper presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013. DOI: 10.1109/ICASSP.2013.6638293.

[6] Kolter, J.Z. and Maloof, M.A., Learning to detect and classify malicious executables in the wild. *The Journal of Machine Learning Research*, 2006. 7: p. 2721-2744.

[7] Christodorescum, M., Jha, S., Seshia, S.A., Song, D. and Bryant, R.E. Semantics-Aware Malware Detection. *SP '05 Proceedings of the 2005 IEEE Symposium on Security and Privacy*, 2005. : p. 32-46.

[8] Kelly, Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. S. Malware Images: Visualization and Automatic Classification. Published in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, 2011. Article No. 4. DOI:10.1145/2016904.2016908.

[9] Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *The Journal of Machine Learning Research, 2003.* 3: p. 1157-1182.

[10] Sak, H., Senior, A., and Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Presented at Interspeech. 2014. : p. 338-342.

[11] Kaggle. Microsoft Microsoft Malware Classification Challenge (BIG 2015). Retrieved from https://www.kaggle.com/c/malware-classification.