Les Cahiers du GERAD ISSN: 0711–2440

Deep learning for proactive cooperative malware detection system

A. Abusitta,
O. Abdel Wahab, T. Halabi

G–2020–23–EIW12

April 2020

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2020
– Bibliothèque et Archives Canada, 2020

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2020
– Library and Archives Canada, 2020

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Deep learning for proactive cooperative malware detection system

Adel Abusitta [a]
Omar Abdel Wahab [b]
Talal Halabi [c]

[a] *McGill Executive Institute, McGill University, Montréal (Québec) Canada, H3A 1G5*

[b] *Université du Québec en Outaouais, Gatineau (Québec) Canada, J8X 3X7*

[c] *Department of Applied Computer Science, University of Winnipeg, Winnipeg (Manitoba), Canada, R3B 2E9*

adel.abusitta@mcgill.ca
omar.abdulwahab@uqo.ca
t.halabi@uwinnipeg.ca

April 2020
Les Cahiers du GERAD
G–2020–23–EIW12

Abstract: *The past few years have seen the ability of cooperative Malware Detection Systems (MDS) to detect complex and unknown malware. In a cooperative setting, an MDS can consult other MDSs about suspicious malware and make a final decision using an aggregation mechanism. However, large delays may arise from both applying an aggregation mechanism and waiting to receive feedback from consulted MDSs. These shortcomings render the decisions produced by existing cooperative MDS approaches ineffective in real-time. To address the above-mentioned problem, we propose a deep learning-based cooperative MDS that efficiently exploits historical feedback data to foster proactive decision-making. More specifically, the proposed approach is based on Denoising Autoencoder (DA), which allows us to learn how to reconstruct complete MDSs' feedback from partial feedback. Our results show the effectiveness of the proposed framework on a real-life dataset.*

1 Introduction

The current communication and computing infrastructure is becoming more and more complex and vulnerable to cyber attacks. In the recent years, studies and results have shown that the use of cooperative Malware Detection Systems (MDSs) can enhance the detection accuracy compared to traditional single MDSs [1, 4, 5]. This is such since it is becoming increasingly difficult for one single MDS to detect all existing malware [1, 2, 5], due to its limited knowledge of such malware patterns and implications. The cooperation among MDSs can be achieved through allowing them to exchange their malware analysis feedback and exploit each other's expertise to cover unknown malware patterns, thus achieving mutual benefits.

There are considerable delays associated with adopting existing cooperative MDS approaches [1, 5]. These delays are mostly due to the computation complexity of using aggregation algorithms such as Bayesian Theory, and also the large geographic distances that separate the MDSs. In fact, each MDS, after receiving feedback from consulted MDSs regarding a suspicious malware, is required to use a suitable feedback algorithm, in order to make a final decision about the suspicious malware. The aggregation method is usually costly in terms of computation time and depends on many factors such as the number of consulted MDSs, and MDSs' trust levels and expertise [1, 2, 5]. In addition, due to the uneven MDSs' connections and communication speeds and other unknown factors (e.g., busy MDSs, compromised MDSs), there is no guarantee that feedback will be synchronously received. Therefore, decisions on whether or not to raise an alarm regarding some suspicious malware might be excessively delayed due to the missing feedback of a single MDS. Hence, the decisions generated by the cooperative MDS are ineffective in a real-time setting.

To address the above-mentioned limitations, we propose a proactive cooperative MDS that integrates a deep learning approach. The proposed approach exploits the historical MDSs' feedback to predict the status of a certain suspicious malware. This is done proactively without having to apply any aggregation mechanism on consulted MDSs' feedback, nor having to wait until receiving all the feedbacks from the consulted MDSs, i.e., only partial and/or incomplete feedback can be used to predict the status of suspicions attack. This, in turn, makes our approach reliable and feasible in real-time environments, where decisions on malware must be rapidly taken in order to effectively apply the required action measures at the right time. More particularly, the proposed approach is based on Stacked Denoising Autoencoders (SDAE), where a denoising autoencoder is used as a building block to train a deep neunetwork [11, 12]. We capitalize on the fact that a denoising autoencoder can learn how to reconstruct original inputs giving partial data inputs, through allowing deep neural networks to learn how to extract features that are robust to incomplete MDSs' feedback. Our contributions are summarized as follows:

- Proposing a cooperative malware detection system that enables decision-making on suspicious malware, even with partial MDSs's feedback.
- Designing a proactive cooperative MDS, which enables us to make decisions about suspicious malware proactively, i.e., without the need to apply aggregation mechanisms on MDSs' feedback.

2 SDAE-MDS: The proposed approach

In this section, we first present the concept of traditional autoencoders. Then, we explain the proposed approach.

2.1 The traditional autoencoders

An autoencoder is an unsupervised learning method that is used to learn reliable data codings [9]. It is used to pre-train each layer in a deep neural network in order to obtain better initial weights that lead to a better-performing classification [3]. Researchers have reported that weights initialization using autoencoders can improve the performance of deep neural networks, compared to a random initialization [3].

An autoencoder is used as a building block for deep networks [3]. In particular, it takes an input vector (MDSs' feedback) $x \in [0,1]^d$, where d is the vector dimension, and maps it to a hidden representation $h \in [0,1]^{d'}$ using the following equation:

$$h = f_\theta(x) = Sig(W * x + b) \tag{1}$$

$\theta = \{W, b\}$, W is a weight matrix and b is a bias vector. Thereafter, the resulting hidden layer representation h will be reconstructed to the output layer x' using a decoding function as follows:

$$x' = g_{\theta'}(h) = Sig(W' * h + b') \tag{2}$$

$\theta' = \{W', b'\}$, W' and b' are a weight matrix and a bias vector of the reverse mapping, respectively. The purpose of the model is to optimize the parameters of the model, so that the reconstruction error between the input and output can be reduced [6].

2.2 The proposed approach

In order to make an autoencoder robust to incomplete MDSs' feedback, the autoencoder should be trained to reconstruct its MDS's feedback even if the feedback does not represent the whole MDSs' feedback (i.e., when some feedback are not available). The autoencoder that deals with corrupted version of the input is called a denoising autoencoder [11]. This is achieved by adding noise to the initial input x before passing it to the hidden layer. The objective is to reconstruct x, where x represents the MDSs' feedback. Thus, a partially corrupted version z will be obtained from x as follows: $z = alpha(x)$ where $alpha$ is a corruption mechanism [11]. In our model, we use Masking Noise Approach (MNA) for the corruption process, as it is useful to represent incomplete MDS's feedback [12]. In MNA noise, a fraction v (selected at random) of each MDSs' feedback x is forced to be 0, while the others remain untouched. In fact, other noise can also be used such as Gaussian noise. However, MNA noise is more useful to simulate incomplete MDSs' feedback [12] since the noise will only change partial feedback.

The autoencoder is then used to take corrupted data z and attempt to learn how to reconstruct x. This is done by allowing the input z to be mapped to a hidden representation, i.e.,

$$h = f_\theta(z) = Sig(W' * z + b') \tag{3}$$

Note that we select z as input instead of x since a traditional autoencoder was used. The value of h is then used to reconstruct x' as follows:

$$x' = g_{\theta'}(h) = Sig(W * h + b) \tag{4}$$

The denoising autoencoder architecture is described in Figure 1. As given in the traditional autoencoder, the parameters are trained to minimize the average reconstruction error:

$$\begin{aligned}\theta^*, \theta'^* &= arg \;\; minimize_{\theta,\theta'} \; \frac{1}{n}\sum_{j=1}^{n} L(z^{(j)}, x'^{(j)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} L(z^{(j)}, g_{\theta'}(f_\theta(z^{(j)})))\end{aligned} \tag{5}$$

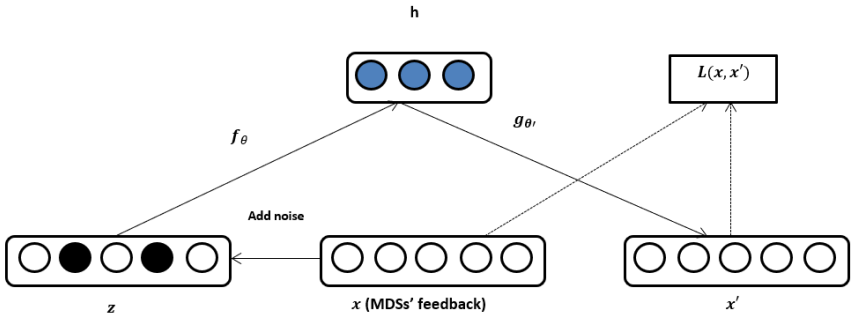

Figure 1: MDS-based denoising autoencoder architecture

The training algorithm of the proposed MDS-based denoising autoencoder is described as follows. For the raw inputs x, we randomly select parts of them to be set to 0 as the corrupted inputs z. The corrupted input z will then be encoded to the hidden code and reconstructed to the output. Note that x' is a deterministic function of z rather than x. The reconstruction that is computed between z and x is denoted as $L(x, x')$. The parameters of the model are randomly initialized and then optimized using stochastic gradient descent algorithms. The above mentioned-steps are performed for each layer added in the proposed MDS-based deep neural network. To generate a classifier for MDS, we add a classifier (e.g., logistic regressions) to the last layer. Then, the parameters of all the layers will be fine-tuned to minimize the error of predicting the target label (i.e., malware or not) using a back-propagation algorithm [3, 7, 8, 11, 12].

3 Evaluation results

To evaluate the proposed model, we create a dataset containing MDSs' feedback on suspicious malware. This dataset was created based on the Android Malware Dataset (MAD) [10], where each 1 or 0 in the new dataset corresponds to the answer of an MDS to a given row of the MAD dataset [10]. The created dataset is used to train the proposed model. Then, the ability of the proposed approach in making decisions about suspicious malware was tested in the presence of partial/incomplete feedback. To represent partial/incomplete MDSs' feedback, some of the MDSs' feedback (selected randomly) were left blank. In this case, blanks indicate that some of MDSs' feedback are yet to be received, due to some unexpected delays (e.g., busy MDS).

The accuracy of the proposed approach is first tested and compared in a complete information scenario, i.e, all MDSs' feedback is received on time. This is useful to evaluate the effectiveness of the proposed approach in making decisions given partial feedback. Figure 2 shows that the accuracy of the proposed model, with a variety of hidden units, was slightly degraded (less than 1.1%). These results suggest that the proposed deep leaning-based approach is able to effectively makes the right decisions on suspicious malware events, even in the presence of some incomplete feedback.

The proposed model (i.e., SDAE-MDS) was also compared with another approach, namely the Stacked Auto Encoder-MDS (SAE-MDS). SAE-MDS uses traditional autoencoders as a building block

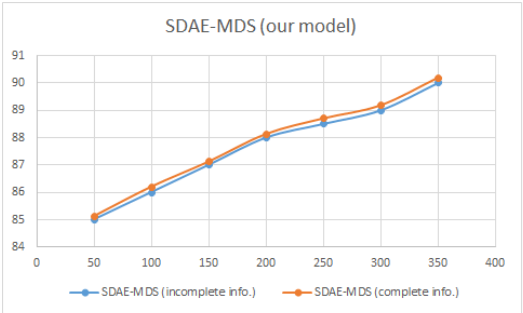

Figure 2: Detection accuracy performance compare to having all the MDSs' feedback (complete information) - number of hidden layers = 3.

for the deep neural networks. The study was conducted with different numbers of layers and hidden nodes. Our model (Figure 3) yields an increased accuracy compared to SAE-MDS (Figure 4). The reason is that we use denoising autoencoders as a building block for our deep neural networks, which allow us to extract robust features that lead to a better classification, despite the incomplete feedback given as inputs to the deep neural network [11]. The denoising autoencoder learned how to reconstruct the feedback from corrupted inputs.

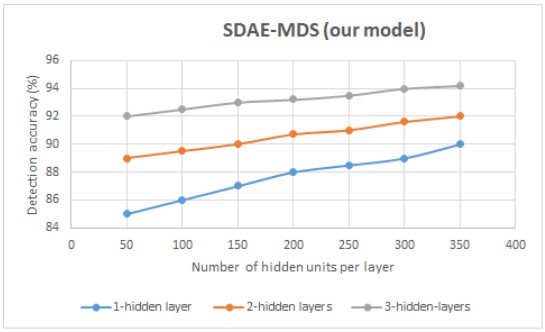

Figure 3: Detection accuracy of SDAE-MDS

Figure 4: Detection accuracy of SAE-MDS

Figure 5 compares SDAE-MDS (the proposed model) with two other denoising models based on training with noisy input, namely SAE(1)-MDS and SAE(2)-MDS. SAE(1)-MDS is a 3-hidden-layers SAE-MDS where noisy inputs were only used for the pretraining. However, SAE(2)-MDS is also 3-hidden-layers SAE-MDS where noisy inputs were used for both pretraining and fine-tuning. The results demonstrate that our model is also resilient to the increase in the percentage of noises.

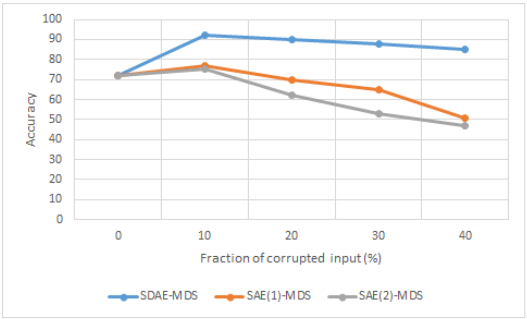

Figure 5: SDAE-MDS vs. training with noisy input

Note that when the corrupted inputs (percentage) equals 0%, all models (SAE(1)-MDS, SAE(2)-MDS and SDAE-MDS) yield the same results in terms of classification accuracy. This is due to the fact that when 0% is applied, the three models will be the same as SAE.

4 Conclusion

We proposed a proactive cooperative MDS. The proposed approach allows us to exploit historical feedback to produce learning models that can effectively and efficiently predict the label (malware or not) of the suspicious malware even when some feedback are missing. The proposed approach is based on stacked denoising autoencoders, where we use a denoising autoencoder as a building block for the deep learning classifier. The proposed MDS-based denoising autoencoder is used to learn how to reconstruct original MDSs' feedback given partial ones. The proposed model can also make decisions regarding suspicious malware without having to apply any aggregation mechanism on the consulted MDSs' feedback. Experimental results show the effectiveness of the proposed approach.

References

[1] Adel Abusitta, Martine Bellaiche, and Michel Dagenais. A trust-based game theoretical model for cooperative intrusion detection in multi-cloud environments. In 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), pages 1–8. IEEE, 2018.

[2] Adel Abusitta, Martine Bellaiche, and Michel Dagenais. Multi-cloud cooperative intrusion detection system: trust and fairness assurance. Annals of Telecommunications, 74(9–10):637–653, 2019.

[3] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In Advances in neural information processing systems, pages 153–160, 2007.

[4] Áine MacDermott, Qi Shi, and Kashif Kifayat. Collaborative intrusion detection in federated cloud environments. Journal of Computer Sciences and Applications, 3(3A):10–20, 2015.

[5] Carol J Fung and Quanyan Zhu. Facid: A trust-based collaborative decision framework for intrusion detection networks. Ad Hoc Networks, 53:17–31, 2016.

[6] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning, volume 1. MIT press Cambridge, 2016.

[7] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554, 2006.

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.

[9] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. Neurocomputing, 139:84–96, 2014.

[10] Laya Taheri, Andi Fitriah Abdul Kadir, and Arash Habibi Lashkari. Extensible android malware detection and family classification using network-flows and api-calls. In 2019 International Carnahan Conference on Security Technology (ICCST), pages 1–8. IEEE, 2019.

[11] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103. ACM, 2008.

[12] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec):3371–3408, 2010.