Uma Proposta para Classificação de Famílias de Programas Maliciosos baseada em Texturas

Tamy E. Beppler[1], **Luiz E. S. Oliveira**[1], **André R. A. Grégio**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19097 – 81531-980 – Curitiba – PR – Brazil

{tebeppler, lesoliveira, gregio}@inf.ufpr.br

Abstract. *Malware classification using texture analysis is an approach that has been widely used by many authors to solve the problem of family categorization. Interpreting a binary file as an image can bring advantages, for example, to avoid recent malware obfuscation techniques. Over the last years, different classifiers and texture descriptors were proposed to achieve higher accuracy rates. In this paper, we executed experiments using Random Forest for classification on a public and a local datasets, showing a discussion about related work inadequate use of datasets to build general malware classifiers.*

Resumo. *Classificação de malware utilizando análise de texturas é uma abordagem que tem sido amplamente empregada por diversos autores para resolver o problema de categorização de famílias. Interpretar um arquivo binário como imagem pode trazer vantagens, por exemplo, evitar as atuais técnicas de ofuscação usadas por malware. Ao longo dos últimos anos, foram propostos diferentes classificadores e descritores de textura para alcançar altas taxas de acurácia. Neste artigo são realizados experimentos utilizando Random Forest para classificação em um dataset público e em um local, apresentando uma discussão sobre o uso de datasets não apropriados pela literatura para construir classificadores genéricos de malware.*

1. Introdução

Ataques a computadores, servidores e outros dispositivos eletrônicos são observados há muitos anos. Arquivos e programas com intenções maliciosas buscam danificar, modificar, excluír, roubar dados e/ou se multiplicar. Ao conseguir identificá-los é possível interrompê-los. Com isso, arquivos ou programas maliciosos (*malware*) tem sido amplamente estudados para compreensão de seu funcionamento e, consequentemente, detecção e mitigação de suas atividades infecciosas. Uma vez identificado que se trata de um arquivo malicioso através da detecção de *malware* (que pode ser realizada usando diferentes técnicas, como visto em [Idika and Mathur 2007]), esses arquivos podem ser categorizados de acordo com sua plataforma, em classes por comportamento apresentado e em famílias por função específica [Nataraj 2015]. Dentro das famílias há diversas variantes com pequenas diferenças que, embora as distinguam, permitem o seu agrupamento. Das várias maneiras possíveis para se classificar um *malware*, uma opção promissora é

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

a análise de texturas, dado que sua aplicação em outros cenários tem apresentado elevadas taxas de sucesso [Costa et al. 2012] [Bertolini et al. 2013] [Cid et al. 2017], além de evitar certas técnicas de ofuscação e funcionar para qualquer sistema operacional. Nela, os binários são convertidos em imagens e um vetor de características pode ser extraído (usando um descritor) para ser utilizado como entrada do classificador.

Neste artigo utiliza-se *Random Forest* com 50 estimadores para classificação de *malware* baseada em texturas atingindo 98.7% de acurácia no *dataset* público, Malimg [Nataraj et al. 2011a], taxa de acerto superior às atingidas pelos outros trabalhos da literatura que usam o mesmo *dataset* completo, sendo esta a primeira contribuição. Adicionalmente, foi realizada uma comparação do *dataset* utilizado na literatura com uma base local de arquivos maliciosos, levantando a um questionamento sobre a qualidade dos resultados já publicados quando da sua aplicação em um *dataset* representativo e sem escolha de amostras. Os experimentos mostram que utilizar amostras selecionadas elevam significativamente a acurácia de classificação, a principal contribuição desta pesquisa.

Na Seção 2, apresenta-se uma introdução sobre análise e classificação de texturas. A Seção 3 retrata o atual estado da arte para classificação de *malware* usando textura. A Seção 4 mostra o método proposto do processo de classificação e na Seção 5 são apresentados os resultados obtidos, bem como uma breve discussão. Por fim, a Seção 6 conclui o presente artigo e mostra os trabalhos futuros.

2. Fundamentação Teórica

Antes do processo de classificação, os arquivos de *malware* devem ser transformados em texturas, para que sejam extraídas suas características. Esta seção traz uma breve explicação sobre a análise de texturas e classificação de *malware*.

Análise de Texturas. A análise dinâmica ou estática de *malware* são as mais comumente utilizadas. No primeiro caso o arquivo é executado em um sistema operacional em geral virtualizado/emulado e gera um relatório comportamental com base no rastreamento da execução, porém demanda mais tempo e recursos, e sofre com técnicas de evasão quando detectado ambiente virtual. No segundo, o código é "desmontado" e explorado em busca de padrões maliciosos, porém sofre com técnicas de ofuscação. Para evitar esse tipo de problema, [Nataraj et al. 2011a] propõe o uso de análise de textura, tratando o arquivo como uma imagem, uma vez que permite capturar padrões normalmente não capturados pelas outras abordagens [Nataraj et al. 2011b]. Em seus experimentos o autor mostra que essa análise não sofre com ofuscação de código ou evasão, é muito mais rápida e funciona para diferentes sistemas operacionais. Para isso, o binário é convertido em uma imagem em escala de cinza e são extraídas e analisadas as características da imagem, por fim utilizadas na classificação dos arquivos. Para extrair as características é usado um descritor de texturas. Os descritores mais utilizados podem ser divididos basicamente em locais e globais. Os globais, como GIST [Oliva and Torralba 2001], analisam a imagem como um todo, atribuindo características que seguem um padrão. Já os descritores locais, como *Local Binary Pattern* (LBP) [Ojala et al. 2002], dividem a imagem em blocos e registram características de blocos da imagem.

Classificação. Atualmente existem muitos classificadores que podem ser utilizados para categorizar famílias de *malware* e muitos já foram utilizados na literatura para análise de texturas, com resultados de até 98.88% de

acurácia [Makandar and Patrot 2016]. Para [Kosmidis 2016], há uma necessidade de maiores estudos em relação a classificação de *malware*, pois cada conjunto de dados é um problema diferente a ser tratado. Assim, em busca de uma proposta com uma boa taxa de acerto e mais genérica quanto ao conjunto de dados, este trabalho usa *Random Forest*. Trata-se de um dos algoritmos para classificação mais utilizados por ser bastante simples: cria-se um conjunto de árvores de decisão e as combina [Breiman 2001]. Trata-se de uma ferramenta efetiva na previsão de classes e inserindo o tipo certo de aleatoriedade faz com que sejam classificadores muito precisos. A implementação do SciKit Learn combina os classificadores fazendo uma média da probabilidade de predição, ao invés de deixar o classificador optar por uma única classe [scikit-learn developers]. Para os experimentos realizados nesta pesquisa foram utilizados 50 estimadores (árvores de decisão).

3. Trabalhos Relacionados

A classificação de *malware* por análise de textura foi proposta primeiramente por [Nataraj et al. 2011b], que analisaram binários de *malware* convertendo-os em textura, extraíram as características em vetores de atributos com o descritor GIST e os classificaram em famílias usando *k-nearest neighbors* (KNN), obtendo taxa de acurácia de $\approx$98%. Os autores compararam diferentes descritores e *datasets*, mostrando que a abordagem funciona em qualquer sistema operacional e não sofre com as técnicas de ofuscação atuais [Nataraj et al. 2011a] [Nataraj 2015] [Nataraj and Manjunath 2016]. [Makandar and Patrot 2015a] propuseram um classificador de *malware* por análise de textura usando, além do GIST, o descritor *Gabor Wavelet Transform* (GWT) e, classificando com *Artificial Neural Networks* (ANN), obtiveram acurácia de 96.35%. Em [Makandar and Patrot 2015b], com o classificador *Support Vector Machine* (SVM), alcançou 86.68%. Para esses foi usada a base de dados Malheur [Rieck et al. 2011].

[Kosmidis and Kalloniatis 2017] estudaram diferentes classificadores para análise de textura de *malware*, onde a melhor taxa de acerto foi de 91.6% de acurácia na base Malimg [Nataraj et al. 2011a] com *Random Forest* e descritor GIST. Em [Makandar and Patrot 2016], os autores começaram a utilizar a *Discret Wavelet Transform* (DWT) junto com o descritor GIST e, ao classificarem usando KNN, obtiveram 98.88% de acurácia. De forma similar, [Makandar and Patrot 2017a] usa KNN (98.84%) e SVM (98.88%), porém com uma etapa extra após o descritor de textura: a seleção de características usando *Principal Component Analysis* (PCA). [Makandar and Patrot 2017b] e [Makandar and Patrot 2018] focaram em classificação de *malware* em famílias dentro da classe Trojan. No primeiro caso, usando os conjuntos de dados Malheur e Malimg, o descritor DWT e classificador SVM, atingiram 91.05% e 92.53% de acurácia respectivamente; já no segundo trabalho, utilizaram GWT com os classificadores KNN e SVM, com resultados de 89.11% e 75.11% no conjunto de dados Malimg. Usando um descritor local, *Local Binary Pattern* (LBP), [Luo and Lo 2017] apresentaram uma melhora na classificação de *malware* quando comparada com GIST. Ao usar os classificadores *Convolutional Neural Networks* (CNN) com TensorFlow, SVM e KNN com o LBP, obtiveram 93.17%, 87.88% e 85.93%, respectivamente, enquanto com GIST atingiram 87.88%, 81.23% e 82.83%.

O uso de redes neurais convolucionais para classificação de famílias de *malware* baseadas em textura foi proposta também por [Yue 2017] que, usando CNN com 43 camadas, atingiu acurácia de 97.32% e 98.63% considerando perda ponderada. [Singh 2017]

também utilizou CNN em uma base com mais de 60.000 exemplares extraídos de alguns repositórios como [Malshare 2012], [VirusShare 2011] e [VirusTotal 2017]. Seu classificador com quatro camadas, sendo duas convolucionais e duas camadas densas, atingiu 95.24% de acurácia, enquanto usando ResNet (*Residual Network*) com sete camadas convolucionais e uma camada *pooling* alcançou 98.21%. [Yakura et al. 2018] compararam o método proposto por [Nataraj et al. 2011a] no conjunto de dados VX Heaven [VXHeaven 2016] usando CNN com um mecanismo que permite gerar um mapa de atenção com as áreas com maior importância para classificação.[Rezende et al. 2017] utilizou *Deep Convolutional Neural Networks* (DCNN) baseado em ResNet com 50 camadas para classificação de *malware*, após a conversão em escala de cinzas seguido por conversão em imagem RGB para alimentar a *Deep Neural Network* (DNN), obtendo acurácia de 98.62%. [Kabanga and Kim 2017] usaram uma CNN de três camadas na base Malimg atingindo 98% de acurácia. Os autores afirmam que, conforme estudos anteriores, "*uma pequena alteração na imagem (que pode não ser visível a olho humano) pode levar a um erro de classificação das imagens*".

4. Metodologia

Para realizar a classificação de *malware* por análise de texturas foi utilizada a mesma metodologia de [Nataraj et al. 2011a], disponível em [Laks 2014], na qual são realizados cinco passos descritos a seguir e ilustrados na Figura 1.

Passo 1 - Converter binários em imagens digitais: cada byte do arquivo binário é representado por um píxel na escala de cinzas (0, preto; 255, branco) e define-se uma largura fixa, enquanto a altura varia conforme o tamanho do arquivo.

Passo 2 - Organizar a base: para posterior classificação, as amostras são separadas de acordo com sua família, obtida com os rótulos do [VirusTotal 2017]. Para o *dataset* local foi utilizado o AVClass [Sebastián et al. 2016] que faz a seleção do rótulo.

Passo 3 - Calcular vetor de características: são extraídas as características das amostras, pelas quais será calculada a semelhança entre as variantes das famílias. Para computar as características, é comum usar operações de redimensionamento, filtragem e média de blocos [Nataraj 2015]. Neste artigo usou-se o descritor GIST para obter as características com um redimensionamento de 128x128, valor atribuido empiricamente que apresentou melhores resultados.

Passo 4 - Classificação supervisionada: é usado um algoritmo de classificação para agrupar as amostras em famílias. Para essa pesquisa foi usado *Random Forest* com 50 estimadores.

Passo 5 - Visualização de resultados: utiliza-se métricas de classificação. Optou-se pela acurácia para comparar a taxa de acerto entre os classificadores e a matriz de confusão, para se observar o resultado obtido para cada família.

5. Testes e Resultados

Nesta seção são apresentados os resultados obtidos após implementar a metodologia proposta. Comparou-se o classificador *Random Forest* com os resultados obtidos na literatura no *dataset* público e realizados novos experimentos em um *dataset* local.

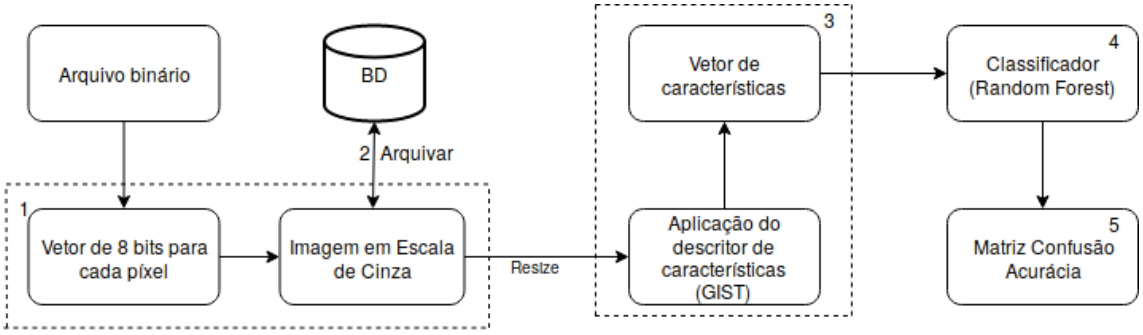

Figura 1. Passos da classificação de malware baseada em análise de texturas.

5.1. *Datasets*

Foi usado o *dataset* Malimg [Nataraj et al. 2011a], que possui 9.342 amostras já convertidas em imagens em escala de cinza de 25 famílias de *malware*. As amostras foram classificadas nessas famílias utilizando o [VirusTotal 2017] e, conforme [Nataraj 2015], os rótulos foram baseados em seis fabricantes de antivírus (AVG, AntiVir, BitDefender, Ikarus, Kaspersky e Microsoft), considerando pelo menos dois similares.

Adicionalmente, para verificar a integridade dos experimentos realizados na literatura, foi utilizado um *dataset* local com 16.598 amostras capturadas desde 2007 na Internet brasileira pelos próprios autores (apresentando um cenário mais próximo ao real). A base também foi rotulada através do [VirusTotal 2017], com o rótulo selecionado pelo AVClass. As amostras foram distribuídas em 706 famílias, todas já convertidos em escala de cinza.

5.2. Implementação

Os experimentos foram feitos com validação cruzada 10-*fold*, pré-processamento das imagens redimensionadas para 128x128 e extração de características com o descritor GIST. Para classificação, optou-se pelo uso do classificador *Random Forest*.

5.3. Visualização de Resultados

Testes com a aplicação do *Random Forest* no *dataset* público resultaram em 98.7% de acurácia. A Tabela 1 compara os resultados dessa proposta com os encontrados na literatura. Isso leva a concluir que utilizar *Random Forest* para *malware* baseado em texturas é um **caminho promissor**.

Na Tabela 1 observa-se que a melhor taxa de acerto é alcançada por [Makandar and Patrot 2016] e [Makandar and Patrot 2017a], porém os autores usaram apenas um subconjunto do *dataset*. Utilizando as mesmas famílias que foram apresentadas nos resultados desses autores, o algoritmo de classificação proposto consegue atingir 100% de acurácia. A Figura 2 apresenta a matriz de confusão gerada usando *Random Forest* no *dataset* público. Na matriz é possível observar que variantes das famílias `Swizzor.gen!I` e `Swizzor.gen!E` são mais semelhantes visualmente, pois são as que mais se confundem entre si. Essas famílias não são consideradas no subconjunto avaliado por [Makandar and Patrot 2016] e [Makandar and Patrot 2017a].

Optou-se também por realizar alguns experimentos em um *dataset* local, com uma maior quantidade de amostras e famílias, para observar o resultado em um cenário mais

Tabela 1. Comparação da acurácia de classificação no *dataset* Malimg com o estado da arte.

Autor e Ano	Acurácia (%)
[Nataraj et al. 2011a]	98.00
[Nataraj et al. 2011b]	98.00
[Nataraj 2015]	98.37
[Kosmidis 2016]	91.60
[Makandar and Patrot 2016]	98.88
[Yue 2017]	98.63
[Luo and Lo 2017]	93.17
[Makandar and Patrot 2017a]	98.88
[Makandar and Patrot 2017b]	92.53
[Rezende et al. 2017]	98.62
[Makandar and Patrot 2018]	89.11
[Kabanga and Kim 2017]	98.00
Proposta	98.70

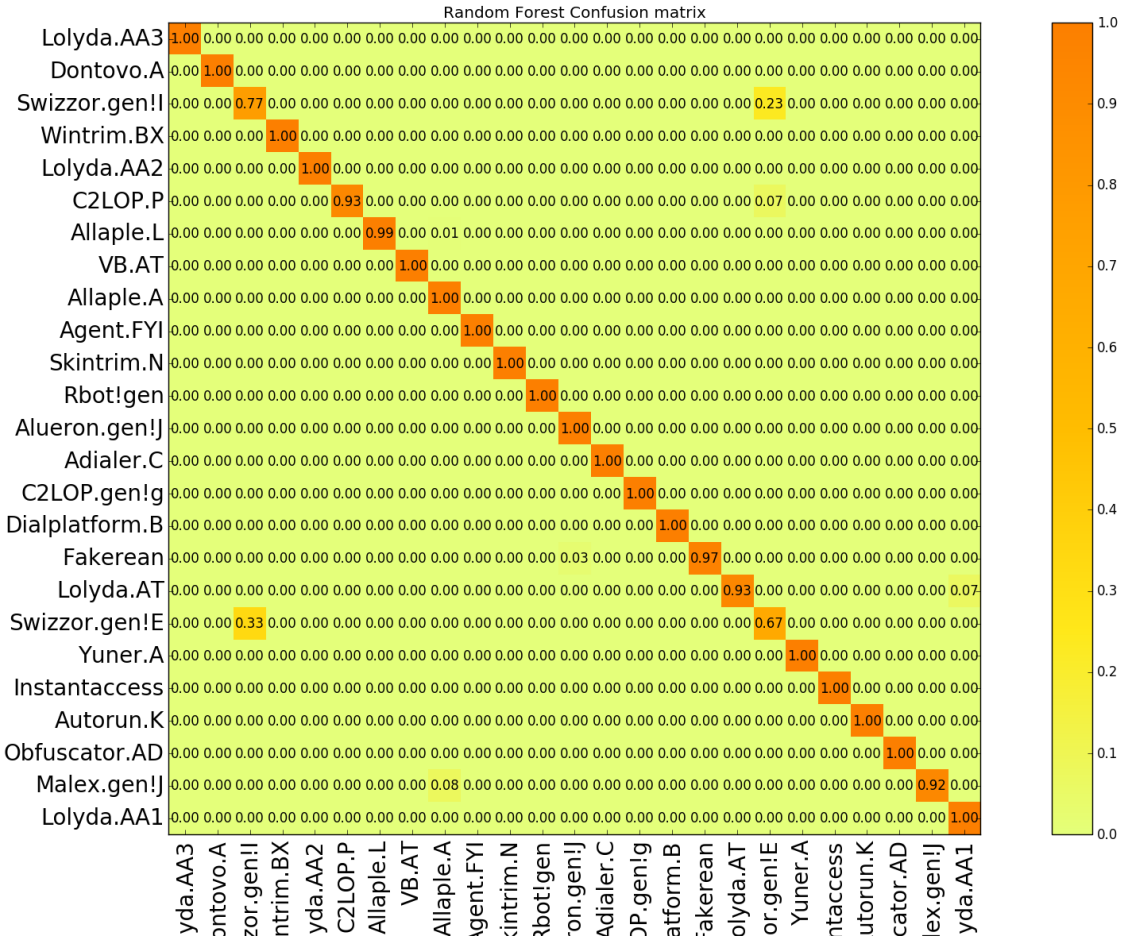

Figura 2. Matriz de confusão do classificador *Random Forest* no *dataset* Malimg.

próximo da realidade. Usando o *dataset* local sem escolha prévia de amostras, observou-se uma taxa de acerto muito pequena. Isso se dá ao fato de possuir uma grande variedade de famílias, bem como a pouca quantidade de amostras em algumas delas. Com poucas amostras os classificadores não possuem tantas características fortes que descrevam uma família. Apesar de ser um resultado bastante desanimador, é o cenário mais próximo da realidade. A Figura 3 mostra os resultados com o *dataset* local usando *Random Forest*.

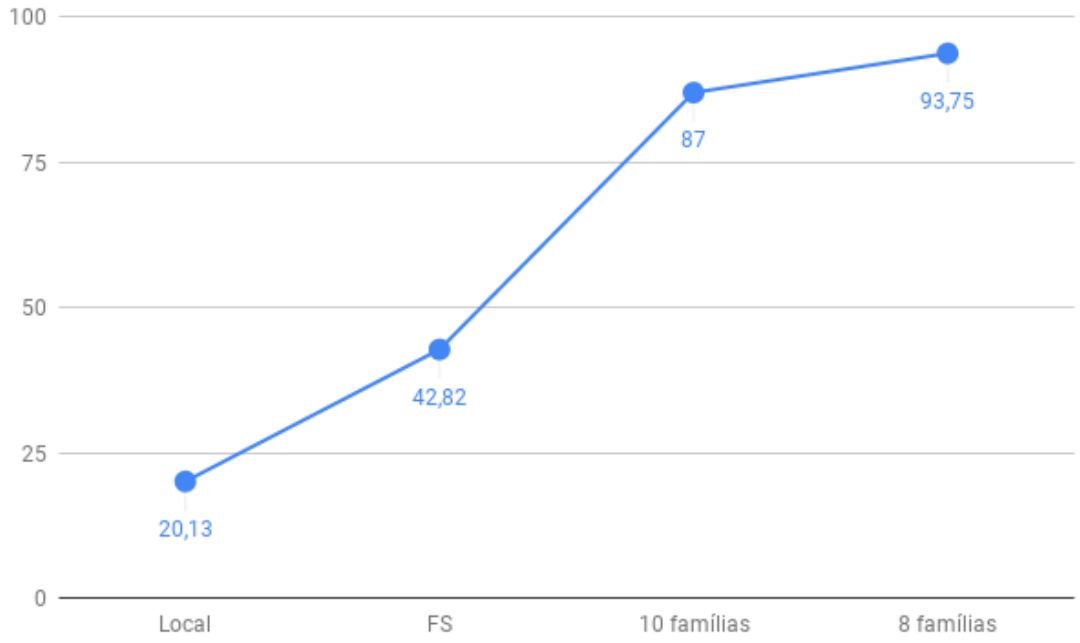

Figura 3. Acurácia de classificação (%) usando *Random Forest* no *dataset* local (FS: famílias mais significativas - com 50 amostras ou mais).

Ao utilizar as 54 famílias mais significativas (FS), pode-se notar uma pequena melhora na classificação, porém os resultados ainda são insatisfatórios. Foram utilizadas 8650 amostras de 54 famílias, comparável com o *dataset* da literatura em quantidade de amostras, mas ainda é uma grande variedade de famílias para serem distinguidas. Dentre essas famílias, foram selecionadas as dez com maior representatividade, utilizando 100 amostras aleatórias de cada uma delas, o que produziu um resultado mais expressivo (87%). Avaliando os resultados com esse conjunto de dados, é possível observar que quando usada uma quantidade menor de famílias, é possível atingir uma taxa de acerto ainda mais elevada. Seguindo a metodologia anteriormente utilizada em [Makandar and Patrot 2016] e [Makandar and Patrot 2017a], foram realizados experimentos com 8 famílias aleatórias usando 100 amostras selecionadas também aleatoriamente, o que resultou numa acurácia de 93.75%.Isso mostra que sob as mesmas condições é possível atingir uma taxa de acerto bastante elevada, apenas fazendo a seleção das famílias a serem classificadas. Isso claramente é uma subversão do problema de classificação no mundo real, onde as entradas para o classificador serão as mais variadas possíveis, incluindo-se aí programas benignos que não devem ser classificados como maliciosos.

6. Considerações Finais

A análise de texturas utilizada na classificação de *malware* já apresenta resultados bastante expressivos quanto a taxa de acerto das famílias. Nesse trabalho, as imagens foram redimensionadas em uma escala de 128x128 e optou-se por usar o descritor GIST. A partir disso, foi utilizado *Random Forest* com 50 estimadores atingindo um resultado ainda mais significativo no *dataset* Malimg. Em um cenário mais próximo ao real, esse tipo de análise não obteve bons resultados. Como visto, os trabalhos da literatura (e o *dataset* público) apresentam resultados que podem estar enviesados, dada a escolha das famílias que apresentam melhores resultados de classificação. Como trabalhos futuros, pretende-se investigar a eficiência do classificador *Random Forest* para classificação de *malware* em tempo real, na detecção de exemplares maliciosos e benignos e na efetividade da generalização de um modelo.

Por fim, para essa pesquisa foi utilizado o descritor mais comumente aplicado (GIST), porém, por ser um descritor global, pode perder algumas características importantes, portanto um dos trabalhos futuros é avaliar a classificação usando um descritor local, como o LBP, que já foi testado na literatura e apresentou bons resultados.

Referências

Bertolini, D., Oliveira, L., Justino, E., and Sabourin, R. (2013). Texture-based descriptors for writer identification and verification. *Expert Systems with Applications*, 40(6):2069 – 2080.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Cid, Y. D., Müller, H., Platon, A., Poletti, P., and Depeursinge, A. (2017). 3d solid texture classification using locally-oriented wavelet transforms. *IEEE Transactions on Image Processing*, 26(4):1899–1910.

Costa, Y., Oliveira, L., Koerich, A., Gouyon, F., and Martins, J. (2012). Music genre classification using lbp textural features. *Signal Processing*, 92(11):2723 – 2737.

Idika, N. and Mathur, A. P. (2007). A survey of malware detection techniques. *Purdue University*, 48.

Kabanga, E. K. and Kim, C. H. (2017). Malware images classification using convolutional neural network. *Journal of Computer and Communications*, 6(01):153–158.

Kosmidis, K. (2016). Machine learning and images for malware detection and classification. Master's thesis, School of Science Technology, Thessaloniki, Greece.

Kosmidis, K. and Kalloniatis, C. (2017). Machine learning and images for malware detection and classification. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, pages 5:1–5:6. ACM.

Laks (2014). Sarvam blog. `http://sarvamblog.blogspot.com.br`. Acessado em 25/08/2017.

Luo, J. and Lo, D. C. (2017). Binary malware image classification using machine learning with local binary pattern. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 4664–4667.

Makandar, A. and Patrot, A. (2015a). Malware analysis and classification using artificial neural network. In *2015 International Conference on Trends in Automation, Communications and Computing Technology*.

Makandar, A. and Patrot, A. (2015b). Malware image analysis and classification using support vector machine. *International Journal of Trends in Computer Science and Engineering*, 4(5):01–03.

Makandar, A. and Patrot, A. (2016). An approach to analysis of malware using supervised learning classification. In *International Conference on Recent Trends in Engineering, Science Technology*.

Makandar, A. and Patrot, A. (2017a). Malware class recognition using image processing techniques. In *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*.

Makandar, A. and Patrot, A. (2017b). Wavelet statistical feature based malware class recognition and classification using supervised learning classifier. *Oriental Journal of Computer Science and Technology*.

Makandar, A. and Patrot, A. (2018). Trojan malware image pattern classification. In *Proceedings of International Conference on Cognition and Recognition*, pages 253–262, Singapore. Springer Singapore.

Malshare (2012). Malware repository. `https://malshare.com/`. Acessado em 30/07/2018.

Nataraj, L. (2015). *A Signal Processing Approach To Malware Analysis*. PhD thesis, University of California, Santa Barbara, CA, USA.

Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. S. (2011a). Malware images: Visualization and automatic classification. In *Proc. 8th International Symposium on Visualization for Cyber Security*.

Nataraj, L. and Manjunath, B. S. (2016). SPAM: signal processing to analyze malware. *CoRR*.

Nataraj, L., Yegneswaran, V., Porras, P., and Zhang, J. (2011b). A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In *Proc. 4th ACM Workshop on AISec*.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(7):971–987.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175.

Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., and de Geus, P. (2017). Malicious software classification using transfer learning of resnet-50 deep neural network. In *2017 16th IEEE ICMLA*.

Rieck, K., Trinius, P., Willems, C., and Holz, T. (2011). Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 19(4):639–668.

scikit-learn developers. *Scikit-learn User Guide*. scikit-learn.

Sebastián, M., Rivera, R., Kotzias, P., and Caballero, J. (2016). Avclass: A tool for massive malware labeling. In *Research in Attacks, Intrusions, and Defenses*, Cham. Springer International Publishing.

Singh, A. (2017). Malware classification using image representation. Master's thesis, Department of Computer Science and Engineering - Indian Institute of Technology Kanpur, Kanpur, UP, India.

VirusShare (2011). Malware repository. `https://virusshare.com/`. Acessado em 30/07/2018.

VirusTotal (2017). Virustotal. `https://www.virustotal.com/`. Acessado em 25/04/2017.

VXHeaven (2016). Vx heaven virus collection. `http://83.133.184.251/virensimulation.org/`. Acessado em 08/08/2018.

Yakura, H., Shinozaki, S., Nishimura, R., Oyama, Y., and Sakuma, J. (2018). Malware analysis of imaged binary samples by convolutional neural network with attention mechanism. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*.

Yue, S. (2017). Imbalanced malware images classification: a CNN based approach. *CoRR*, abs/1708.08042.