Perquirere, 17 (2): 241-254, maio/ago. 2020. *ISSN*: 1806-6399

https://revistas.unipam.edu.br/index.php/perquirere

Inteligência artificial aplicada à segurança da informação

Artificial intelligence applied to information security

Alex Bruno Gonçalves Almeida
Graduando do curso de Sistemas de Informação (UNIPAM)
E-mail: alexbga@unipam.edu.br

Juliana Lilis da Silva
Professora orientadora (UNIPAM)
E-mail: juliana@unipam.edu.br

Resumo: Este artigo aborda a aplicação de técnicas da inteligência artificial (IA) na segurança da informação, visando à identificação de arquivos executáveis maliciosos e legítimos. Para comprovar a eficiência da aplicação, é apresentada uma análise da capacidade e desempenho de detecção do conteúdo malicioso em relação às ferramentas utilizadas no mercado atual. No desenvolvimento do trabalho, foi empregada a linguagem de programação *Python* e técnicas de aprendizado de máquina.
Palavras-chave: *Malware*. Inteligência Artificial. Aprendizado de Máquina. Segurança da Informação.

Abstract: This article addresses the application of artificial intelligence (AI) techniques in information security, aiming to identify malicious and legitimate executable files. To prove the efficiency of the application, an analysis of the capacity and performance of detection of malicious content is presented in relation to the tools used in the current market. In the development of the work, Python programming language and machine learning techniques were used.
Keywords: Malware. Artificial Intelligence. Machine Learning. Information Security.

1 INTRODUÇÃO

Nos últimos anos, a tecnologia teve um enorme avanço em todo o mundo. A cada dia, surgem novidades nas mais diversas áreas de atuação, seja corporativa, seja comercial, seja educacional, que irão facilitar a vida dos usuários. Quando se fala em tecnologia, já se tem em mente uma grande quantidade de dados que estão sempre envolvidos. Tais dados podem estar expostos a ameaças, acidentais ou intencionais, a roubos ou até mesmo a modificações externas.

Percebe-se também que a quantidade de usuários que utilizam a internet vem crescendo cada vez mais, porém o Instituto Brasileiro de Geografia e Estatística (IBGE)[1]

[1] Disponível em: https://ibge.gov.br/.

apresenta que "um total de 63,4 milhões de pessoas com faixa etária superior aos 10 anos de idade informaram não usar a *web*. Destes, 37,8% disseram que não sabem usar e outros 37,6% que não se interessam pelo recurso" (PNAD, 2018). Essas informações podem ser usadas, a princípio, para se pensar na utilização da internet e como sua má utilização pode agravar fatores como a segurança da informação.

Outro ponto a ser observado é que atualmente empresas estão investindo mais na segurança da informação devido à grande quantidade de ataques cibernéticos que têm ocorrido nos últimos tempos, sendo o crime cibernético a maior ameaça à sobrevivência das organizações (ALLEN, 2013). Os crimes cibernéticos afetam a segurança dos dados, por isso, para qualquer negócio, a proteção, a longo prazo, acaba sendo extremamente positiva.

Diante de tantos fatores críticos a serem preservados, há uma necessidade de uma proteção que possa controlar o tráfego de redes em diferentes níveis de confiança, que possa racionalizar o uso dos recursos de rede e mitigar riscos que podem ser expostos.

Nesse contexto, a proposta desse trabalho foi empregar a inteligência artificial (IA) interligada à segurança da informação, realizando comparações entre ferramentas de segurança convencionais de mercado com uma ferramenta de segurança que utiliza IA. O objetivo foi comprovar que essa ferramenta é capaz de identificar se o objeto analisado apresenta sinais de risco ao usuário. Ela pode ainda atuar na detecção de falsos positivos, para que assim o usuário tenha uma melhor tomada de decisão sobre o objeto presente.

2 REFERENCIAL TEÓRICO

Essa seção visa a expor conceitos abordados no trabalho.

2.1 SEGURANÇA DA INFORMAÇÃO

A segurança de informação é caracterizada pela aplicação adequada de dispositivos de proteção sobre um ativo (ou um conjunto de ativos), visando a preservar o valor que este possui para as organizações. A aplicação dessas proteções busca preservar a Confidencialidade, a Integridade e a Disponibilidade (CID), não estando restritas somente a sistemas ou aplicativos, mas também a informações armazenadas ou veiculadas em diversos meios além do eletrônico ou em papel (BASTOS; COUBIT, 2009).

Quando se fala em segurança da informação, refere-se a tomar ações para garantir a confidencialidade, a integridade, a disponibilidade e os demais aspectos da segurança das informações dentro das necessidades do cliente (LYRA, 2008). Esses três principais pilares estão definidos a seguir:

- Confidencialidade: "Garantia de que o acesso à informação é restrito aos seus usuários legítimos." (BEAL, 2008, p. 1). Ou seja, seu acesso é permitido apenas a determinados usuários.

- Integridade: "Toda informação deve ser mantida na mesma condição em que foi disponibilizada pelo seu proprietário, visando a protegê-las contra alterações indevidas, intencionais ou acidentais" (SÊMOLA, 2003, p. 45). Ou seja, informação não adulterada.
- Disponibilidade: "Garantia de que a informação e os ativos associados estejam disponíveis para os usuários legítimos de forma oportuna" (BEAL, 2008, p. 1). Ou seja, independentemente da finalidade, a informação deve estar disponível.

Na segurança da informação, o rompimento de qualquer um desses pilares pode trazer enormes efeitos negativos a uma organização, como a parada inteira da linha de produção até a total exposição de dados confidenciais.

2.2 ANTIVÍRUS

Desempenhando o papel de um vigia vinte e quatro horas, que monitora todas as atividades de um computador, um antivírus é um software que detecta, impede e atua na remoção de softwares maliciosos (CANALTECH, 2014), que podem surgir por meio de sites de conteúdo duvidoso, e-mails, pendrives, *download* de arquivos entre diversos modos.

Para ocorrer as detecções dos conteúdos maliciosos, os antivírus rastreiam informações e comparam o comportamento dos novos arquivos com as informações que já estão guardadas em suas bases de dados (VLCEK, 2015). Devido a essa função, as empresas de antivírus sempre buscam manter sua base interna de dados atualizada, coletando assinaturas que servem como impressões digitais dos vírus (VLCEK, 2015), o que faz com que sempre tenham que correr atrás dos criadores de conteúdos maliciosos.

2.3 INTELIGÊNCIA ARTIFICIAL

Sendo uma das ciências mais recentes, a IA teve seu início após a segunda Guerra Mundial, atuando em áreas como aprendizado e percepção e até em tarefas específicas como jogos de xadrez. A IA sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Nesse sentido, ela é um campo universal (RUSSELL; NORVIG, 2004).

Na ciência da computação, a IA é definida como a capacidade de um sistema interpretar, aprender e usar dados para atingir resultados antes não alcançados (KAPLAN; HAENLEIN, 2019). Surgiu na década de 50, com o objetivo de desenvolver sistemas para resoluções de problemas e simular eventos do ser humano, como pensar.

Na IA, percebe-se o processamento de símbolos da computação a fim de se encontrarem métodos genéricos para automatizar atividades perceptivas e cognitivas, por meio de um computador (PEREIRA, *s.d.*), podendo inclusive comportar aspectos de psicanálise e aspectos de psicossíntese.

2.4 APRENDIZADO DE MÁQUINA

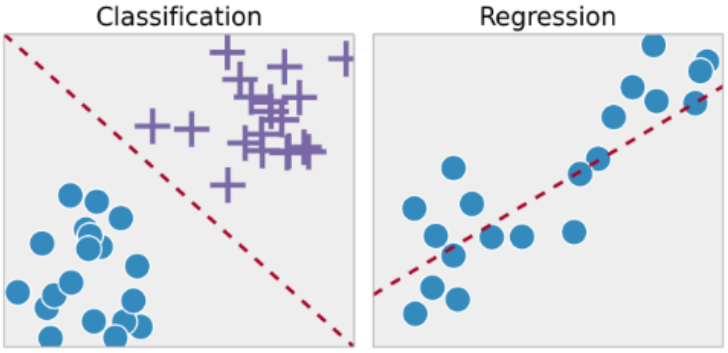
Sendo parte de um dos ramos da inteligência artificial, aprendizado de máquina, ou *Machine Leaning* (ML), relaciona-se à construção de programas, que automaticamente melhoram com sua experiência (MITCHELL, 1997). É perceptível a importância do aprendizado de máquina à medida que a computação vai evoluindo. O aprendizado de máquina trouxe grandes inovações ao mundo. Nota-se a grande movimentação em relação aos carros autônomos e até em recomendações de filmes e músicas que uma pessoa possa consumir.

Isso ocorre devido à grande facilidade de se criarem algoritmos que possam coletar dados e aprender com eles e consequente realizar uma predição sobre alguma coisa no mundo (COPELAND, 2016). Alguns métodos de aprendizado são utilizados para se chegar ao resultado esperado, como a classificação supervisionada.

2.4.1 Aprendizado Supervisionado

Algoritmos aplicados com aprendizado supervisionado possuem a capacidade de aplicar o que foi aprendido no passado a novos dados, usando exemplos rotulados para prever eventos futuros (EXPERT SYSTEM, 2017). Nesse tipo de aprendizado, há a possibilidade de se aplicarem pesos ou de se calibrarem níveis de assertividade e de precisão de um modelo. São divididos em dois grupos que, são representados na Figura 1.

Figura 1 – Exemplo de Classificação e Regressão



Fonte: BARROS, 2016.

Os grupos apresentados na Figura 1 são definidos a seguir:

- Classificação: adiciona-se uma entrada e atribui-se um rótulo a ela, então são métodos usados com objetivo de respostas binárias (BARROS, 2016); é usado neste trabalho com a capacidade de prever se um arquivo executável é malicioso ou legítimo, por meio das características recebidas desse arquivo.

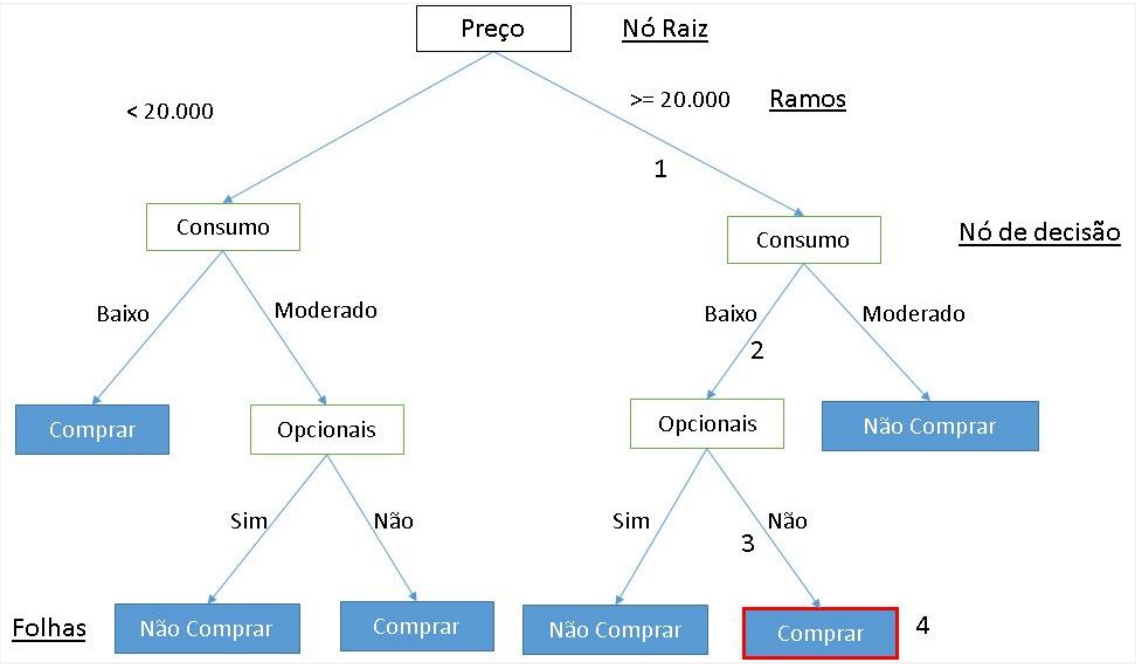
- Regressão: são métodos usados com previsões não binárias (BARROS, 2016); são aplicados a modelos de perguntas como "Qual idade?" ou "Quanto custa?", ou seja, dada a imagem de um homem ou mulher, o modelo teria capacidade de prever a idade dessa pessoa.

E ainda podem existir algoritmos que utilizam o método misto, ou seja, podem utilizar partes de regressão para fazer a classificação ou o oposto.

2.4.1.1 Árvores de Decisão

Extremamente utilizada em problemas de classificação, árvore de decisão é uma ferramenta de suporte à decisão, que usa gráfico ou modelo de decisão baseado em uma árvore (BRID, 2018). Nas árvores de decisão, cada nó interno representa um "teste" em um atributo, cada ramo representa o resultado do teste, cada nó da folha atribui uma classificação e, por fim, caminhos da raiz representam as regras de classificação (DEVMEDIA, 2014), como representados na Figura 2.

Figura 2 – Exemplo de Classificação e Regressão



Fonte: DEVMEDIA, 2014.

A ideia básica de qualquer algoritmo de árvore de decisão é, incialmente, selecionar o melhor atributo, torná-lo um nó de decisão e dividir o conjunto de dados em subconjuntos menores. Inicia-se a construção da árvore repetindo esse processo recursivamente para cada filho até que as condições de parada correspondam à não existência de mais instâncias e atributos restantes, e todas as tuplas pertencerem ao mesmo valor de atributo (NAVLANI, 2018).

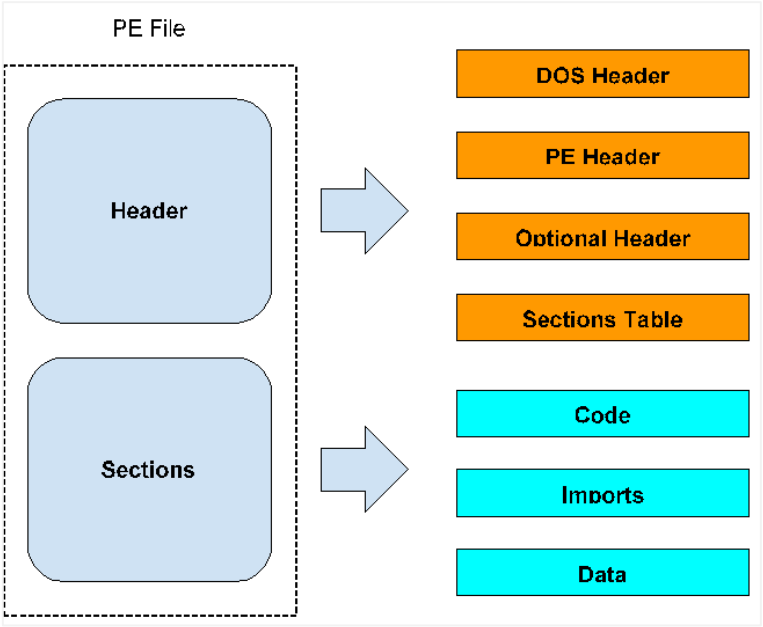
2.5 *WINDOWS PORTABLE EXECUTABLE* (PE)

Sendo o espelho de um sistema operacional, o formato do *Portable Executable* (PE) tem um papel fundamental em todos os sistemas operacionais da Microsoft (PIETREK, 1994). É o formato que possui a estrutura padrão de armazenamento de dados de arquivos, onde são encapsuladas informações como código, dados de inicialização, sequência de execução, checagem de compatibilidade de versão do sistema operacional, entre várias outras informações.

O formato PE é composto por cabeçalhos e sessões, sendo que cada um deles possui outras características importantes para compor totalmente a estrutura de um arquivo PE. Essas aracterísticas são presentes em arquivos com a extensão *exe*, *dll* e *sys*. Observando a Figura 3, pode-se entender um pouco sobre como é a estrutura citada.

O *Header* ou cabeçalhos, como o nome sugere, consiste na sequência de *bits* que descrevem os dados no início do arquivo (WIKIBOOKS, 2018). É o setor onde se encontram a assinatura do cabeçalho PE, bibliotecas que serão usadas para execução e funcionamento do arquivo executável, versão do sistema operacional, descrição de como a memória deverá ser tratada entre diversas informações (WIKIBOOKS, 2018).

Figura 3 – Estrutura de um arquivo PE.



Fonte: Elaborado pelos autores, 2019.

Conforme mostra a Figura 3, *Sections* ou sessões é o local onde estão salvos instruções, recursos, dados e informações do arquivo, sendo também onde se encontra o código do arquivo executável (DEVMEDIA, 2005). A Figura 4 representa como é um pouco do cabeçalho PE. Nesse caso, foi utilizando *Python* 2.7 para capturar as características.

Figura 4 – Captura de informações do PE.

```
1 #Biblioteca usada para capturar arquivos PE
2 import pefile
3 #Arquivo de exemplo a ser usado para coletar os exemplos
4 exe_path = "/home/alexbruno/TCC/MalwareDetect/exe-dir/legitimate/atmlib.dll"
5 #Capturando todas as informacoes PE do arquivo selecionado
6 pe = pefile.PE(exe_path)
7 #Mostrando na tela apenas as caracteristicas selecionadas abaixo
8 print("[*] Machine value: %s" % pe.FILE_HEADER.Machine)
9 print("[*] e_magic value: %s" % hex(pe.DOS_HEADER.e_magic))
10 print("[*] Signature value: %s" % hex(pe.NT_HEADERS.Signature))
```

Fonte: Elaborado pelo autor, 2019.

No código, é usada a biblioteca *Pefile*, e o arquivo usado como exemplo para se terem as características PE adquiridas foi o ATMLIB.DLL. Assim, na linha 6 são atribuídas todas as características desse arquivo para a variável "pe". Como resultado do código, características como valor de máquina, número mágico e valor da assinatura são mostradas na tela. Mas há formas de se mostrarem todas as características PE deste arquivo, trazendo a possibilidade de o usuário final analisá-lo como bem entender, sendo passível inclusive de modificações de características, dependendo do nível de conhecimento do usuário.

3 METODOLOGIA

O presente estudo teve natureza descritiva, de caráter bibliográfico, com intuito de se adquirir a capacidade de decidir se a utilização de ferramentas de segurança aplicadas à IA obteria um melhor desempenho e segurança do que ferramentas usuais que o mercado pode oferecer.

Então foi desenvolvido um algoritmo de aprendizado de máquina, utilizando-se a linguagem de programação *Python*, o qual é capaz de pegar o conteúdo PE de um arquivo executável e, com base em uma classificação realizada por esse algoritmo e na seleção das melhores características, definir se um arquivo é malicioso ou legítimo.

Para validação do algoritmo de aprendizado de máquina, foram utilizados treze mil e quinze arquivos, sendo dez mil maliciosos, retirados do acesso ao site Virus Share, que é um repositório de arquivos maliciosos disponibilizado para a comunidade. O restante foi retirado de uma máquina com sistema operacional recém-instalado sem suas devidas atualizações e sem conexão com a internet, garantindo maior precisão no conteúdo legítimo.

Então foram realizadas a montagem de 24 máquinas virtuais, sendo 6 para cada sistema operacional escolhido: Windows 7, 8, 10 e Server 2016. Foram também definidas configurações de máquina idênticas, com o propósito de se realizarem análises das detecções de *malwares*, garantindo-se melhor segurança, precisão dos dados e desempenho.

Pretendeu-se, portanto, obter informações palpáveis de cenários em que pudesse haver investimento da IA aplicada à segurança da informação e a cenários em

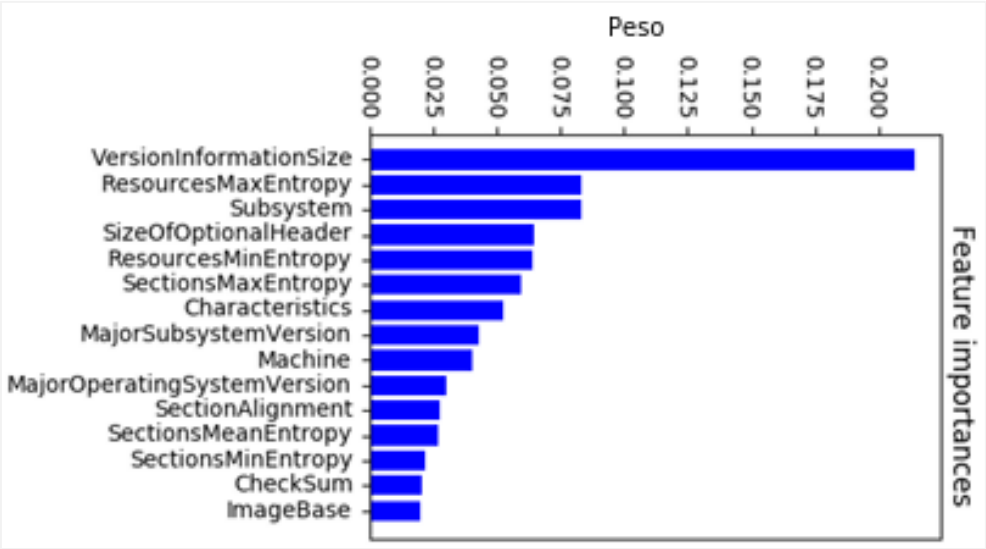
que os resultados pudessem trazer melhor desempenho após a utilização da inteligência artificial.

4 DESENVOLVIMENTO E RESULTADOS

Para o desenvolvimento do trabalho, inicialmente foram realizadas pesquisas sobre o funcionamento das técnicas *maching learnig* a fim de se encontrar o melhor método para criação da solução. Foi utilizada a técnica de aprendizado supervisionado, sendo o projeto divido em etapas.

Na primeira delas, foi realizada a coleta de dados, sendo determinada a utilização da base de dados com características do formato PE contendo 20.955 arquivos, os quais foram divididos ao meio entre conteúdos maliciosos e legítimos. Posteriormente, para preparação dos dados, foram retirados valores em branco e colunas não importantes ao desenvolvimento da análise. Para se obterem as melhores classificações, foi utilizada a ideia de seleção de *features*, com o propósito de se reduzir o conjunto de recursos para seleção das mais relevantes na identificação. Então, utilizando a seleção de *features*, com base em árvore, obteve-se a identificação de 15 *features* importantes nessa base de dados, conforme mostra a Figura 5.

Figura 5 – Features selecionadas.



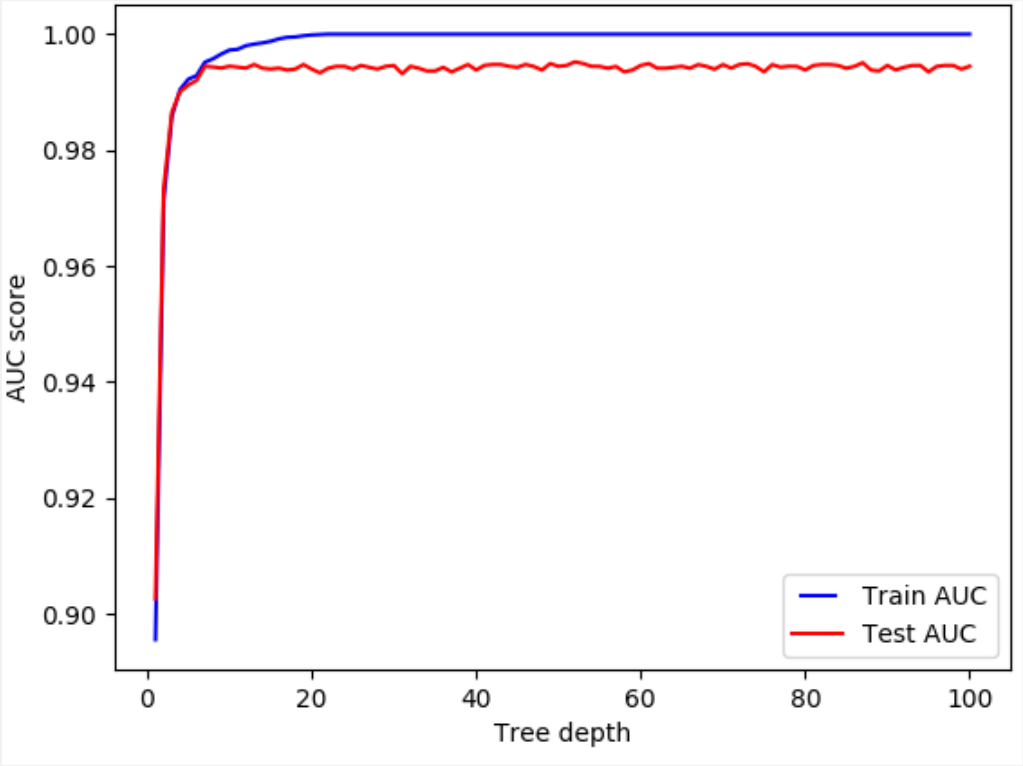
Fonte: Dados da pesquisa, 2019.

Para identificação do arquivo, foi utilizado o modelo binário junto ao algoritmo de árvore de decisão. O modelo binário foi escolhido devido à base de dados apresentar respostas binárias.

Com a utilização da árvore de decisão, houve a possibilidade de melhorar os pesos de cada função dentro do algoritmo. Então foram criados gráficos para mostrar o resultado de cada função. Em seguida, foram selecionados os melhores valores e inseridos em uma função que faz a classificação utilizando todos os pesos adicionados,

buscando-se, assim, a melhor porcentagem possível na classificação. A Figura 6 mostra o resultado da função *tree_depth.*

Figura 6 – Porcentagem adquirida em relação ao número escolhido



Fonte: Dados da pesquisa, 2019.

Após os gráficos plotados, é utilizada a função *GridSearchCV,* como mostra a Figura 7, que trata de um processo de executar o ajuste de hiperparâmetros para determinar os valores ideais para um determinado modelo. Nesse caso, o modelo passado foi o de árvore de decisão, junto aos valores adquiridos nos gráficos anteriores, assim é atribuído à variável "clf" o resultado dos melhores valores especificados.

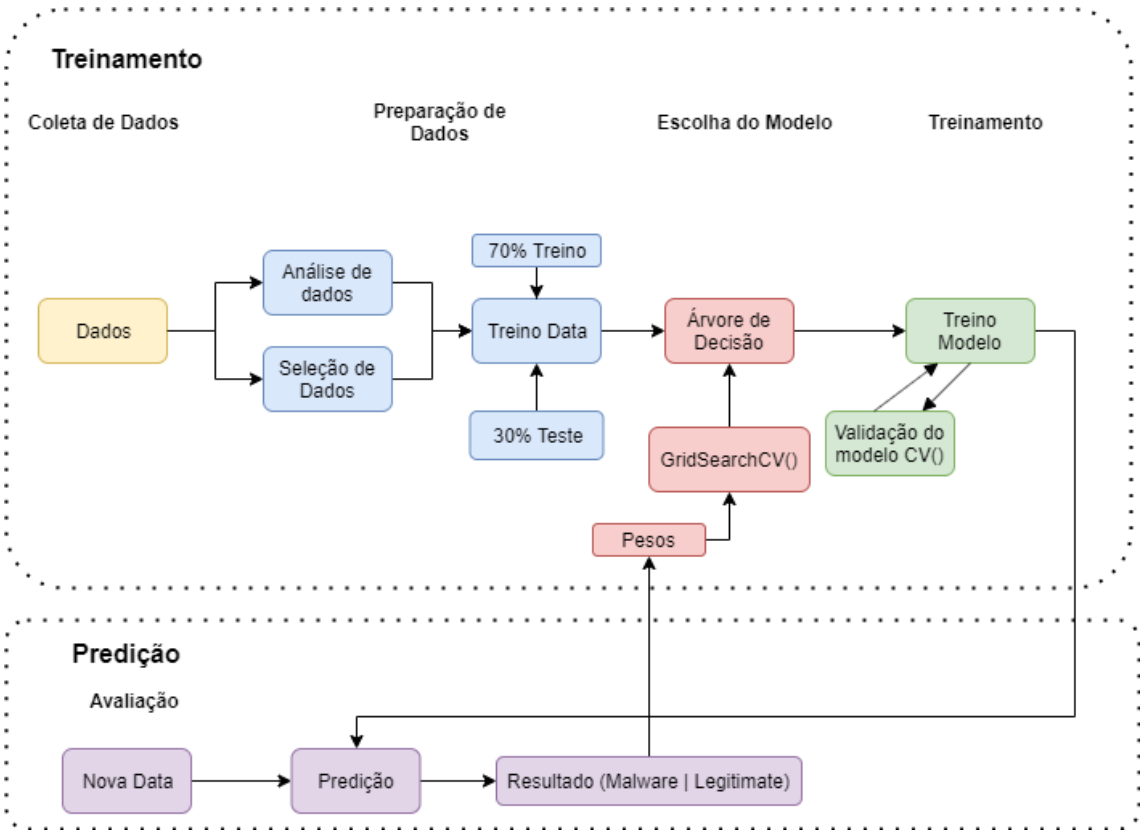
Figura 7 – Porcentagem adquirida em relação ao número escolhido

```
177 #Modelo a user usado: Arvore de Decisão
178 model=DecisionTreeClassifier(random_state=1)
179 #ABaixo todo os parametros selecionados
180 params = {'criterion':['entropy', 'gini'],
181           'max_features': ['auto', 'sqrt', 'log2'],
182           'max_depth':[50,55,60,65,70,75,80,85,90,95,100],
183           'min_samples_split': [5,6,7,8,9,10,11,12,13,14,15],
184           'random_state':[1]}
185 #Teste de todos os parametros
186 clf = GridSearchCV(model, param_grid=params, cv=3, scoring='accuracy')
```

Fonte: Dados da pesquisa, 2019.

Portanto, com os pesos selecionados, o algoritmo é finalmente treinado com uma acurácia acima de 99%. Para avaliação de precisão da classificação, foram selecionados 100 arquivos, em que 50 são legítimos e 50 maliciosos. Nesse caso, há um algoritmo de ML que coleta todas as informações do formato PE do arquivo e, com base nessas informações, identifica se o arquivo é malicioso ou não. Caso o resultado dos 100 arquivos não seja próximo da realidade, ou seja, se gerarem muitos falsos positivos, retorna-se à parte de escolha do modelo, sendo alterados os pesos de cada função.

Figura 8 – Topologia do desenvolvimento



Fonte: Elaborado pelos autores, 2019.

A fim de se finalizar o projeto, foi iniciada a parte de testes das ferramentas. Foi realizada a criação das máquinas virtuais, sendo utilizada a versão gratuita do WMware Workstation, que trata de um software para criação de máquinas virtuais. Foram criadas 6 máquinas para os sistemas operacionais *Windows 7 Home*, *Windows 8.1*, *Windows 10* e *Windows Server 2016*, sendo que cada máquina recebeu uma ferramenta. Para se realizarem as análises, foram escolhidas 5 ferramentas tradicionais no mercado na área de segurança da informação, sendo elas AVAST, AVG, ESET, NORTON e PC PROTECT.

Para os testes serem justos e passíveis de serem comparados, foram adicionados alguns critérios:

1. uma ferramenta por máquina virtual;
2. todas as máquinas devem seguir as seguintes configurações:
   a. 1 Processador;

b. 4 Gb de memória;
3. cada ferramenta utilizará sua melhor versão gratuita;
4. cada ferramenta não poderá fazer a checagem nos arquivos de forma recursiva;
5. cada ferramenta não pode fazer nenhuma varredura inicial no sistema operacional;
6. cada ferramenta só realiza checagem na pasta que conterá os arquivos.

Para se realizarem esses testes, foram selecionados arquivos maliciosos e legítimos. O critério usado para os arquivos legítimos foi que todos os arquivos deveriam ser dos sistemas operacionais usados nos testes, garantido, assim, que todos os arquivos selecionados fossem legítimos. Para selecionar os arquivos, foram usados os comandos apresentados na Figura 9.

Figura 9 – Comando para filtrar arquivos e o inserir dentro de outra pasta

```
1  Get-ChildItem -Path "DIRETÓRIO ORIGEM\*" -Include *.exe,*.dll
        -Recurse | Copy-Item -Destination D:\DIRETÓRIO DESTINO
```

Fonte: Elaborado pelos autores, 2019.

Como resultado, 3.015 arquivos legítimos foram selecionados dos 4 sistemas operacionais. Para a seleção dos arquivos maliciosos, foi requerido ao site VirusShare, que consiste em um repositório de arquivos maliciosos disponíveis ao público, o acesso do autor deste texto. Foi preciso, para tal fim, a demonstração de motivo de necessidade, e o cumprimento das normas impostas aos membros da plataforma em análise.

Após o acesso liberado, foram baixados 10.000 mil arquivos maliciosos da página, com a utilização do comando *CURL*, do sistema operacional Linux.

E, por fim, foram realizados testes em cada ferramenta para se identificarem o tempo gasto e a capacidade para se detectarem os arquivos maliciosos e legítimos, sendo identificados os resultados apresentados nas Figuras 10 e 11.

Figura 10 – Análise em relação a detecção de Arquivos Maliciosos

Malware Detect - 10.000 Files												
SO / Software	Windows 7 Home			Windows 8			Windows 10			Windows Sever 2016		
	Detecção	Tempo	Previsão	Detecção	Tempo	Previsão	Detecção	Tempo	Previsão	Detecção	Tempo	Previsão
IA (MY CODE)	9.553	02:38:48	95,53%	9.551	02:40:59	95,51%	9.516	03:10:59	95,16%	9.553	02:22:17	95,53%
AVAST	6.657	00:51:35	66,57%	8.149	01:09:23	81,49%	7.990	00:47:39	79,90%	NOT SUPPORTED		
AVG	6.539	01:04:38	65,39%	4.448	00:19:52	44,48%	7.772	00:51:11	77,72%	NOT SUPPORTED		
ESET	2.545	00:27:01	25,45%	2.543	00:22:24	25,43%	2.545	00:26:26	25,45%	NOT SUPPORTED		
PC PROTECT	8.902	00:04:43	89,02%	8.900	00:05:35	89,00%	8.901	00:06:50	89,01%	8.902	00:06:56	89,02%
NORTON	NOT SUPPORTED			1.329	01:52:23	13,29%	1.492	04:24:54	14,92%	NOT SUPPORTED		

Fonte: Dados da pesquisa, 2019.

Figura 11 – Análise em relação a detecção de Arquivos Legítimos

Legitimate Detect - 3.015 Files												
SO / Software	Windows 7 Home			Windows 8			Windows 10			Windows Sever 2016		
	Detecção	Tempo	Previsão	Detecção	Tempo	Previsão	Detecção	Tempo	Previsão	Detecção	Tempo	Previsão
IA (MY CODE)	2.893	00:49:06	95,95%	2.893	00:47:40	95,95%	2.893	01:00:07	95,95%	2.893	00:44:31	95,95%
AVAST	3.015	00:01:49	100,00%	3.015	00:00:25	100,00%	3.015	00:00:28	100,00%	NOT SUPPORTED		
AVG	3.015	00:29:34	100,00%	3.015	00:00:58	100,00%	3.015	00:00:41	100,00%	NOT SUPPORTED		
ESET	3.015	00:00:18	100,00%	3.015	00:00:30	100,00%	3.015	00:02:42	100,00%	NOT SUPPORTED		
PC PROTECT	3.015	00:00:24	100,00%	3.015	00:00:33	100,00%	3.015	00:00:49	100,00%	3.015	00:00:33	100,00%
NORTON	NOT SUPPORTED			3.015	00:03:03	100,00%	3.015	00:01:24	100,00%	NOT SUPPORTED		

Fonte: Dados da pesquisa, 2019.

Pela análise das tabelas, é perceptível a alta disponibilidade da utilização do algoritmo com IA, pois as demais ferramentas não possuem suporte para máquinas que utilizem Windows Server. Nota-se também uma alta taxa de detecção com a utilização do algoritmo com IA, mas, devido à metodologia usada, que coleta características de cada arquivo, é notório um tempo maior para realizar essa detecção.

Salienta-se que todas as ferramentas comerciais testadas utilizam das assinaturas dos arquivos maliciosos a fim de se realizar a devida detecção. Por conseguinte, foi notória uma baixa taxa de constatação pelos antivírus analisados, devido à não atualização de suas bases com as assinaturas dos arquivos rodados, tendo como resultado a não obtenção da detecção de mais de 90% dos arquivos. Conclui-se que, caso surja um arquivo malicioso no presente, a capacidade dos antivírus de detectá-lo é inferior, quando comparada à solução que utiliza algoritmo com IA.

5 CONCLUSÃO

O projeto em questão aborda um amplo conteúdo nas utilizações de IA empregando ML, além de expor o vasto campo abordado pela linguagem P*ython*, que pode chegar, especificamente, para esse uso. Somado a tal fato, percebe-se o quão amplo pode ser o uso dessas ferramentas em quaisquer atividades de realização desejável.

Notou-se, na realização desse projeto, que o uso de ML para a identificação de *malware* traz resultados satisfatórios quando se trata de demonstrar, em números, a quantidade de identificações; no entanto, perde no quesito tempo, sendo, muito disso, devido ao método utilizado para realizar tais reconhecimentos, tendo em vista que, quanto maior o arquivo, maior é o tempo para o êxito de sua verificação.

As ferramentas de antivírus realizam sua identificação empregando, basicamente, a assinatura de cada arquivo. Tal fato demonstra que, se o antivírus não possuir essa assinatura em sua base de dados, o arquivo, ainda que malicioso, pode tornar-se algo legítimo para o usuário e, sendo assim, expô-lo a diversos riscos, dependendo do nível que esse *malware* possa exibir.

Ainda é perceptível, quando se trata de ML, a existência de diversos meios que possam apresentar um aperfeiçoamento, sendo eles tanto na identificação, quanto no tempo. O intuito do projeto consistiu em realizar a pesquisa e em desenvolver um algoritmo que seja capaz de aprender com os meios que são impostos a ele e que possua

a capacidade de englobar um aprimoramento na segurança para desfrute do usuário final, quando este possa vir a necessitar.

REFERÊNCIAS

ALLEN, Ken. **Crimes cibernéticos são a maior ameaça à sobrevivência das empresas, aponta estudo da EY**. 2013. Disponível em: https://www.ey.com/br/pt/services/release_pesquisa_seguranca_informacao_ey.

BARROS, Pedro. **Aprendizagem de máquina**: supervisionada ou não supervisionada?. 7 abr. 2016. Disponível em: https://medium.com/opensanca/aprendizagem-de-maquina-supervisionada-ou-não-supervisionada-7d01f78cd80a.

BASTOS Alberto; CAUBIT, Rosângela. **Gestão de Segurança da Informação**: ISO 27001 e 27002: uma visão prática. Porto Alegre: Zouk, 2009.

BEAL, Adriana. **Segurança da informação:** Princípios e melhores práticas para a proteção dos ativos de informação nas organizações. São Paulo: Atlas, 2008.

BRID, Rajesh S. **Decision trees**: a simple way to visualize a decision. 2018. Disponível em: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb.

CANALTECH. **O que é antivírus?** 2014. Disponível em: https://canaltech.com.br/antivirus/o-que-e-antivirus/.

COPELAND, Michael. **What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning**. 2016. Disponível em: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

DEVMEDIA. **Mineração de dados com árvores de decisão**. 2014. Disponível em: https://www.devmedia.com.br/mineracao-de-dados-com-arvores-de-decisao/31397.

DEVMEDIA. **Windows Portable Executable**, 2005. Disponível em: https://www.devmedia.com.br/windows-portable-executable/857.

EXPERT SYSTEM**. What is Machine Learning?:** a definition. 2017. Disponível em: https://www.expertsystem.com/machine-learning-definition/.

KAPLAN, Andreas; HAENLEIN, Michael. **Siri, Siri, in my hand**: who's the fairest in the land?: on the interpretations, illustrations, and implications of artificial intelligence. 2019. Disponível em: https://www.sciencedirect.com/science/article/pii/S0007681318301393.

LYRA, Mauricio Rocha. **Segurança e auditoria em sistemas da informação.** Rio de Janeiro: Ciência Moderna, 2008.

MITCHELL, Tom M. **Machine Learning**. New York: McGraw–Hill, 1997.

NAVLANI, Avinash. **Decision Tree Classification in Python**. 2018. Disponível em: https://www.datacamp.com/community/tutorials/decision-tree-classification-python.

PEREIRA, Luís Moniz. **Inteligência artificial**: mito e ciência. Disponível em: http://www.egov.ufsc.br/portal/sites/default/files/anexos/6511-6510-1-PB.pdf.

PIETREK, Matt. **Peering inside the PE**: a tour of the Win32 Portable Executable File Format. 1994. Disponível em: http://bytepointer.com/resources/pietrek_peering_inside_pe.htm.

PNAD. **Contínua TIC 2016**: 94,2% das pessoas que utilizaram a Internet o fizeram para trocar mensagens. 21 fev. 2018. Disponível em: https://agenciadenoticias.ibge.gov.br/20073-pnad-continua-tic-2016-94-2-das-pessoas-que-utilizaram-a-internet-o-fizeram-para-trocar-mensagens.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 2. ed. Rio de Janeiro: Campos, 2004.

SÊMOLA, Marcos. **Gestão da Segurança da Informação**: uma visão executiva. Rio de Janeiro: Elsevier Campus, 2003.

VLCEK, Ondrej. Como funciona um antivírus? **Canaltech**, 07 dez. 2015. Entrevista concedida a Igor Lopes. Disponível em: https://www.youtube.com/watch?v=g8cqm4-nhYA.

WIKIBOOKS. **Windows Executable Files**. 2018. Disponível em: https:/en.wikibooks.org/wiki/X86_Disassembly/Windows_Executable_Files#Section_Table.