Detecção de Domínios Maliciosos

Abner F. B. Costa[1], **Michele Venturin**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba – PR – Brasil

afbcosta@inf.ufpr.br, mventurin@inf.ufpr.br

Resumo. *Este artigo tem como objetivo apresentar um relatório parcial do projeto prático da disciplina Ciência de Dados para Segurança, ofertada pela Universidade Federal do Paraná (UFPR) e lecionada pelo professor doutor André Ricardo Abed Grégio. Este relatório contém uma breve introdução ao conjunto de dados, uma visão geral do conjunto escolhido abordando aspectos como distribuição dos dados, classes, atributos e características, por fim é apresentado um planejamento futuro do projeto que descreve quais algoritmos de aprendizado de máquina serão testados e quais etapas serão seguidas posteriormente.*

1. Introdução

A rede mundial de computadores, que também recebe o nome de Internet, foi criada em 1969 e desde então vem sendo regulamentada e estudada na tentativa de se tornar cada vez mais segura. No entanto, sua popularização também veio seguida de um grande número de tentativas maliciosas que buscam enganar usuários para obter informações e dados pessoais ou mesmo comprometer dispositivos digitais.

Muitas destas tentativas são realizadas através de nomes de domínios, e podem atuar de maneiras distintas para atingir seus objetivos maliciosos, como fazer download automático de um *malware* quando acessado, solicitar informações do usuário, imitar um site famoso para que o usuário coloque seus dados voluntariamente ou então ser uma fonte de *spam*.

Através de um conjunto de dados de nomes de domínios[Mahdavifar et al. 2021], que foram previamente rotulados como *malware*, *phishing*, *spam* ou benignos, nosso objetivo é construir um algoritmo capaz de classificar corretamente um nome de domínio entre duas categorias benigno ou malicioso. No artigo original proposto, foram geradas características lexicográficas, características estatísticas e características obtidas por meio de terceiros, nossa proposta também incluir reduzir os grupos de características de modo a remover características estatísticas, que foram obtidas na seção de resposta da resposta do DNS.

O principal propósito para construir um algoritmo de classificação para este problema, é a falta de algoritmos robustos capazes de lidar com novos domínios não catalogados, pois abordagens que utilizam uma lista negra tem dificuldade para listar os diversos domínios que são criados a cada dia. Obtendo o algoritmo de classificação desejado, este pode ser usado para auxiliar usuários a identificar ameaças, que muitas vezes são mascaradas colocando nomes similares a nomes de domínios famosos, o diferencial desta aplicação é que como não será necessário fazer requisição alguma ao domínio desejado, adicionando uma camada extra de proteção ao usuário.

2. Visão Geral do Dataset

2.1. Coleta e Pré-Processamento

Os dados coletados foram obtidos através do conjuntos de dados *CIC-Bell-DNS 2021 Dataset* [Mahdavifar et al. 2021], este conjunto contém dados de domínios previamente agrupados em classes.

Para gerar os dados utilizados no experimento, foi realizada uma limpeza no conjunto de dados brutos, convertendo URLs em domínios, removendo duplicatas e integrando diferentes conjuntos de dados em um único contendo as classes utilizadas.

A principio será realizada uma tentativa de classificar todos os domínios, mesmo que estes estejam inativos ou não seja possível obter informações de terceiros.

2.2. Classes e Distribuição

Os domínios foram rotulados entre quatro classes secundárias:

- **Malware:** Domínios identificados que geram qualquer tipo de malware, como *drive-by download*, ataques de negação de serviço distribuída (DDoS) e spyware.
- **Phishing:** Domínios que imitam a aparência de sites legítimos e utilizam técnicas de engenharia social para induzir os usuários a clicar no link malicioso.
- **Spam:** Domínios que empregam buscam encontrar endereços de e-mail válidos para enviar e-mails em massa.
- **Benigno:** Domínios que não pertencia a nenhuma das três classes maliciosas *malware*, *phishing* e *spam*.

A partir destas subclasses foram geradas duas classes principais:

- **Benigno:** Utilizando a classe secundária **Benigno**
- **Não Benigno:** Combinando as classes secundarias **Malware**, **Phishing** e **Spam**.

Originalmente o conjunto de dados possuía 988299 domínios benignos que foram reduzidos para 73418 domínios, visando um melhor balanceamento das classes e uma redução no tempo de processamento, o que resultou na seguinte distribuição do conjunto de dados:

Table 1. Distribuição Definida

Classe	Quantidade	Porcentagem
Benigno	73418	66.67%
Malware	26795	24.33%
Phishing	9078	8.24%
Spam	836	0.76%
`Total`	110127	100.00%

2.3. Atributos

Utilizando as bibliotecas *IPy* e *tld*, para Python3, foram gerados os seguintes atributos para os domínios:

- Formato IP ou não IP.

- (SSD) *Subdomain+Second-Level Domain*
- (SUB) *Subdomain*
- (SLD) *Second-Level Domain*
- (TLD) *Top-Level Domain*

Para casos onde o domínio fornecido era um IP, foram definidos valores padrões para os atributos SSD, SUB, SLD, TDL. Para casos onde a biblioteca *tld* não foi capaz realizar o separação do domínio, este foi atribuído na íntegra ao atributo SLD.

2.4. Características

Através dos atributos foram gerados dois tipos principais de características, características lexicográficas e características obtidas por terceiros. Estes dois tipos características contemplam um total de 24 características geradas.

Table 2. Características Lexicográficas

Nome	Característica	Tipo
C1.1	TLD	*String*
C1.2	Tamanho SSD	*Integer*
C1.3	Tamanho SUB	*Integer*
C1.4	Tamanho SLD	*Integer*
C1.5	Porcentagem de Dígitos SSD	*Float*
C1.6	Porcentagem de Dígitos SUB	*Float*
C1.7	Porcentagem de Dígitos SLD	*Float*
C1.8	No. de Caracteres Não Alfanuméricos SSD	*Integer*
C1.9	No. de Caracteres Não Alfanuméricos SUB	*Integer*
C1.10	No. de Caracteres Não Alfanuméricos SLD	*Integer*
C1.11	Entropia de Shannon SSD	*Float*
C1.12	Entropia de Shannon SUB	*Float*
C1.13	Entropia de Shannon SLD	*Float*
C1.14	No. de SUB	*String*

Table 3. Características de Terceiros

Nome	Característica	Tipo
C2.1	Conexão Estabelecida	*Boolean*
C2.2	Possui Nome Registrado	*Boolean*
C2.3	Possui Endereço	*Boolean*
C2.4	Número de E-mails Registrados	*Integer*
C2.5	País	*String*
C2.6	Estado	*String*
C2.7	Ano de Criação	*Integer*
C2.8	Idade do Domínio	*Integer*
C2.9	Menor Distância do SLD para SLD Famosos	*Integer*
C2.10	Número de SLD Famosos no SUB	*Integer*

As características geradas por terceiros foram obtidas através das bibliotecas *whois*, *ipwhois* e domínios famosos foram definidos utilizando o ranking Alexa. Já a métrica de distancia utilizada foi a métrica de distância de Levenshtein.

3. Planejamento futuro

A partir da definição do problema, decidimos optar uma abordagem de classificação dos dados, que serão particionados em um conjunto de 70% para treino e 30% teste.

O principal algoritmo que será testado será o algoritmo de florestas randômicas, tendo em vista que este algoritmo é capaz de utilizar diferentes tipos de dados para aprendizagem e classificação, como palavras e números, que estão presentes no vetor de características. Um outro fator que nos levou a escolha deste algoritmo é sua capacidade de não ser muito penalizado caso o conjunto de dados não seja balanceado. Além disso, o algoritmo de florestas randômicas é capaz de nos oferecer uma lista de características mais relevantes, possibilitando que novas análises sejam realizadas sobre os resultados obtidos.

Outros dois algoritmos que pretendemos testar são o *K-Nearest Neighbors* (KNN) e o *Multi Layer Perceptron* (MLP), uma vez que bons resultados foram obtidos utilizando KNN [Mahdavifar et al. 2021] e que o MLP pode encontrar limiares de decisão não lineares, entretanto para a utilização destes algoritmos iremos adaptar os vetores de características para valores numéricos e, se for necessário, normalizá-los.

Através de um classificador que consiga identificar se um domínio é benigno ou não, ele pode ser utilizado por usuários como medida de prevenção antes de acessar o domínio desejado. Esta medida de segurança não necessita contato com o domínio em questão, em vista que as características são obtidas lexicograficamente ou por meio de terceiros, preservando a segurança do usuário.

O processo final será dividido nas seguintes etapas:

1. O usuário insere no programa o URL que deseja acessar.
2. Através do URL é obtido o domínio.
3. A partir do domínio, um outro algoritmo gera o respectivo vetor de características.
4. O vetor de características é inserido no modelo de aprendizado de máquina, retornando a classificação.
5. O usuário é informado com a classificação final, a qual pode conter também uma probabilidade ou valor para referência.

O projeto terá continuidade de acordo com o *pipeline* apresentado na Figura 1. Na programação em questão serão testados os classificadores enunciados realizando ajustes nos atributos e características, assim como um aperfeiçoamento dos algoritmos por meio de ajustes de parâmetros, isso em um ciclo contínuo até a obtenção de resultados satisfatórios ou o descarte do classificador em questão. As duas últimas etapas são de comparação e resultados dos classificadores e integração com os algoritmos de pré-processamento, geração de atributos e geração características.

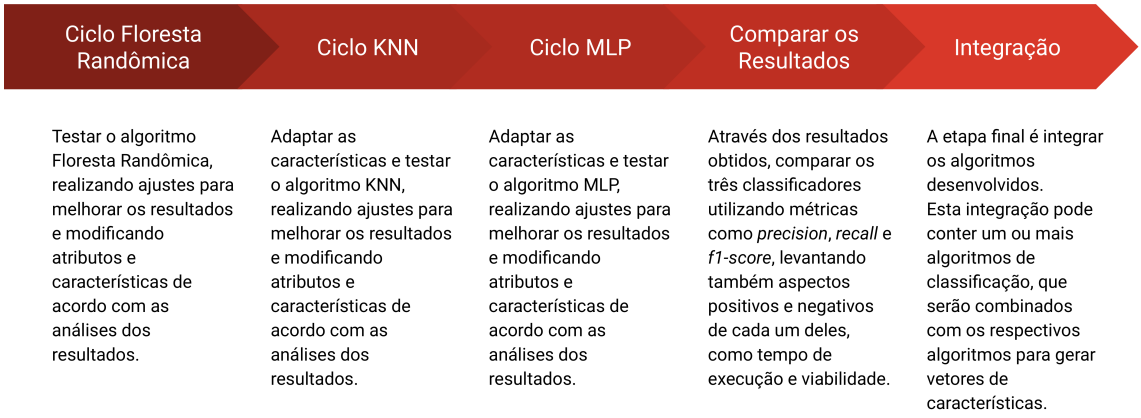

Figure 1. *Pipeline* de execução do Projeto

Após a execução do *pipeline* apresentado será realizada um nova coleta de dados, buscando mesclar novos conjuntos de dados ao conjunto de dados *CIC-Bell-DNS 2021*. Desta forma poderemos comparar os resultados obtidos de nossa abordagem com os resultados originais apresentados e observar se com o conjunto de dados apresentados podemos construir algoritmos que são adequados para outros nomes de domínios, que foram coletados utilizando outras abordagens.

4. Referências

References

Mahdavifar, S., Maleki, N., Lashkari, A. H., Broda, M., and Razavi, A. H. (2021). Classifying malicious domains using dns traffic analysis. *The 19th IEEE International Conference on Dependable, Autonomic, and Secure Computing (DASC)*.