U.C. Introdução à Investigação

Phishing

Docente: Luís Rato
Discentes: André Baião 48092
Gonçalo Barradas 48402
Guilherme Grilo 48921

Março 2022

Resumo

Sendo um dos grandes problemas da atualidade o *phishing*, é uma das formas mais eficazes e utilizadas de crimes cibernéticos, sendo utilizada contra utilizadores individuais, empresas e agências corporativas ou governamentais. O número de ataques registados tem vindo a aumentar, devido principalmente ao aumento da utilização da internet de forma generalizada, havendo cada vez mais informação confidencial e acesso através da internet a dados que são aliciantes aos phishers, como por exemplo o acesso a contas bancarias. O *phishing* tem um elevado impacto económico e social, assim como psicológico.

Existem vários tipos de *phishing*, e várias formas de o combater. As tentativas de combate têm vindo a aumentar, e o desenvolvimento de ferramentas que ajudam a diminuir o número de ataques bem como a sua eficácia também. O objetivo deste trabalho é fazer uma revisão das metodologias que estão implementadas e as que estão a ser desenvolvidas para combater o *phishing*. Fazendo ainda uma revisão dos tipos de *phishing* que existem, bem como os tipos de técnicas que podem ser utilizadas para o combater.

1 Introdução

Atualmente a utilização da internet é generalizada e essencial ao nosso dia-a-dia, havendo por isso cada vez mais partilha de informação online por parte do utilizador. A maioria dos serviços que eram realizados de forma presencial, são atualmente tratados através da internet, havendo dados confidenciais guardados em servidores e disponíveis através da internet, tais como os serviços de finanças, segurança social, bancos, entre outros.

Como resultado desta enorme partilha de informações e transações financeiras existe uma maior vulnerabilidade ao cibercrime. O *phishing* é uma das formas mais eficazes e utilizadas de crimes cibernéticos, sendo utilizada contra utilizadores individuais, empresas e agências corporativas ou governamentais [2].

Nos últimos tempos temos assistido a cada vez mais fraudes online, ataques a sistemas de grandes empresas, bem como acesso a informações confidenciais de empresas, bancos etc,.. existem casos bastante mediáticos um pouco por todo o mundo que incluem este tipo de crimes, sendo por isso essencial arranjar soluções para proteger tanto as empresas como a população em geral deste género de ataques.

O *phishing* pode conter vários tipos de crimes/objetivos, como falsificação de documentos, falsificação de dados informáticos, acesso ilegítimo, burla etc... O impacto económico do *phishing* pode ser avaliado no caso das burlas, mas também em todos os gastos associados à resolução do ataque bem como à prevenção. O impacto social, ou psicológico pode dar origem a desconfiança constante, medo e ansiedade por parte da população. O impacto emocional resultante de um ataque *phishing* pode ser devastador para a vida da vítima, visto que vê a sua privacidade invadida, podendo ocorrer perdas monetárias de elevado valor ou até mesmo vir a ter problemas com a justiça devido à ocorrência de roubo de identidade [4].

2 Estado de arte

Tem havido cada vez mais um esforço por parte dos investigadores, principalmente da área de engenharia informática para desenvolver e criar métodos/modelos que sejam capazes de detetar tentativas *phishing*, antes que seja possível a ocorrência de danos para pessoas ou empresas. Nos últimos anos a maioria destes modelos têm por base a inteligência artificial, visto que é cada vez mais uma área de interesse e em desenvolvimento, com bastantes provas de que consegue ser mais eficaz do que os métodos tradicionais.

Um dos métodos para avaliar um modelo, ou seja, se o modelo é ou não bom para a deteção/classificação é o cálculo da accuracy que é efetuado com base nos resultados do modelo na fase de teste através da seguinte formula:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

Onde: TP – true positive (verdadeiros positivos, ou seja, os dados que foram classificados como positivos e são realmente positivos); TN – true negative (verdadeiros negativos, ou seja, os dados que foram classificados como negativos e são realmente negativos); FN – false negative (falsos negativos, ou seja, os dados que foram classificados como negativos mas na realidade são positivos) e os FP – false positive (falsos positivos, ou seja, os dados que foram classificados como positivos, mas na realidade são negativos).

Apresentamos algumas das ferramentas e métodos anti-*phishing* que foram desenvolvidos nos últimos anos:

Chou et al. (2004) propuseram uma extensão para o browser que funciona com heurística baseada em regras. O SpoofGuard examina várias características dos sites visitados pelo utilizador e com base nessas características calcula a probabilidade de estar perante um site *phishing*, avisando o utilizador se essa probabilidade exceder um valor definido pelo mesmo [6].

Kirda Kruegel (2006) criaram o AntiPhish, que é uma extensão para o browser, que rastreia as informações confidenciais de um utilizador e gera avisos sempre que este tenta fornecer essas informações a um site considerado não confiável [14].

Zhang et al. (2007) criaram uma ferramenta (CANTINA) para deteção de *phishing* com base num algoritmo de aprendizagem (TF-IDF) que compara e classifica as informações presentes na URL [30].

Ronda et al. (2008) desenvolveram o iTrustPage, uma extensão para o browser que funciona através de uma lista branca que contém os sites que já estão classificados como seguros, paralelamente existe uma lista de permissões do próprio utilizador que é atualizada sempre que o utilizador entra num novo

site que seja classificado como seguro [24].

Fahmy Ghoneim. (2011) desenvolveram o PhishBlock, que é uma ferramenta com uma abordagem híbrida para deteção de sites *phishing*, com uma accuracy de 95% [7].

Jeeva Rajsingh (2016) desenvolveram um sistema que efetua a extração dos recurso de uma URL e esses recursos são sujeitos a uma mineração de regras para gerar as regras que definem quais as características que permitem distinguir uma URL legítima de uma que seja *phishing* [11].

Xiao et all (2021), desenvolveram uma rede neuronal convolucional com self-attention, para identificação de URLs de *phishing*, com uma accuracy de 95,6% [29].

Balogun et al. (2021), propuseram vários modelos que utilizam meta-aprendizagem baseados em árvores funcionais para detetar sites *phishing*, obtendo uma accuracy de 98,15% na deteção [5].

Minocha Singh, (2022) desenvolveram um sistema de deteção de *phishing* que utiliza o AV-BMEO, que é uma versão binária do Modified Equilibrium Optimizer (MEO) que utiliza a função de transferência AV e a aprendizagem baseada em oposição para a capacidade de exploração do MEO binário. Em duas análises com dados distintos os autores obtiveram uma accuracy de 96.36 e 97.46% [17].

Alshehri et al. (2022) desenvolveram uma abordagem que utiliza métodos de aprendizagem profunda, nomeadamente um modelo de deep learning para deteção de sites *phishing*. Os autores obtiveram uma accuracy de 98,13% [3].

3 Tipos de *phishing*

Existem vários tipos de *phishing* que são classificados consoante o métodos utilizados, no entanto normalmente um ataque de *phishing* utiliza vários métodos em simultâneo de forma a ser mais efetivo.

3.1 Deceptive Attack

É o tipo mais comum de ataque, no qual são utilizadas técnicas de engenharia social para enganar as vítimas. Este tipo de *phishing* pode ser utilizado através de e-mails, sites, telefone ou redes sociais.

3.1.1 E-mail *phishing*

Um e-mail *phishing* é um e-mail falsificado, que tem como remetente uma pessoa ou instituição em que o destinatário confia, com o objetivo de o convencer e divulgar as suas informações confidenciais. Por vezes este tipo de e-mails são enviados para um grupo de indivíduos que estão associados a uma instituição ou que fazem parte de um mesmo grupo social de forma a que o e-mail seja mais credível, neste caso tem o nome de spear *phishing*. No caso dos destinatários serem personalidades de relevância ou com altos cargos como CEOs por exemplo, tem o nome de whaling. No clone *phishing*, ocorre a clonagem de um e-mail anteriormente recebido pelo destinatário, com alteração de links ou anexos do e-mail [28].

3.1.2 Sites *phishing*

Os sites *phishing* tem uma aparência bastante semelhante ao site legítimo. O utilizador é redirecionado para este site depois de clicar num link que pode estar incorporado num e-mail ou através de um anúncio (clickjacking).[5] [27]

3.1.3 *phishing* telefónico (Vishing e SMishing)

O *phishing* telefónico é realizado através de telefonemas ou mensagens de texto. O utilizador pode receber uma mensagem de alerta de segurança de um banco por exemplo, sendo coagido a efetuar uma chamada, enviar uma mensagem ou entrar num link onde partilha informações confidenciais [1].

3.1.4 *phishing* Social

O *phishing* social inclui o sequestro de contas, ataques de personificação, golpes e distribuição de malware [10].

3.2 Technical Subterfuge

Esta estratégia tem por base o download de um código malicioso no sistema de forma a permitir que o phisher tenha acesso a informações confidenciais. Este tipo de *phishing* pode ser classificado em *phishing* baseado em malware, *phishing* baseado em DNS (Pharming), *phishing* de injeção de conteúdo, Man-in-the-middle *phishing*, Mecanismo de pesquisa *phishing* e Ataques URL.

3.2.1 *phishing* baseado em malware

Este tipo de *phishing* baseia-se na execução de um software malicioso na máquina do utilizador. O malware é descarregado através de truques de engenharia social ou através de vulnerabilidades no sistema de segurança. O *phishing* baseado em malware pode ser de vários tipos (Key Loggers and Screen Loogers, Vírus e Worms, Spyware (sofware de espionagem), Adware, Ransomware, Rootkits, Hosts File Poisoning, Ataques de reconfiguração do sistema).

3.2.1.1 Key Loggers and Screen Loggers

São um tipo de malware geralmente instalado através de e-mails do tipo cavalo de troia ou através de download direto. Este tipo de software monitoriza dados e regista teclas do utilizador, conseguindo assim capturar informações confidenciais relacionadas às vítimas, como nomes, endereços, senhas e outros dados confidenciais [19].

3.2.1.1.1 Vírus e Worms

Um vírus é um pedaço de código que se espalha dentro de uma aplicação ou programa, fazendo cópias de si mesmo de forma autónoma. Os worms são semelhantes aos vírus, mas diferem na forma de execução, pois os worms são executados explorando a vulnerabilidade dos sistemas operativos sem a necessidade de modificar outro programa. Os vírus transferem-se de um computador para outro com o documento ao qual estão conectados, enquanto os worms fazem a transferência através do arquivo de host infectado [12].

3.2.1.1.2 Spyware (sofware de espionagem)

O software de espionagem é um código malicioso projetado para rastrear os sites visitados pelos utilizadores, com o objetivo de roubar informações confidenciais. O Spyware pode ser recebido através de e-mail e após ser instalado no computador, assume o controlo sobre o dispositivo e altera as suas configurações, tendo a capacidade de recolher informações como senhas e números de cartão de crédito ou registos bancários que podem ser usados para roubo de identidade [21].

3.2.1.1.3 Adware

O adware é um tipo de malware que mostra ao utilizador uma janela pop-up sem fim com anúncios que podem prejudicar o desempenho do dispositivo. Alguns dos adwares podem ser usados para rastrear os sites da internet que o utilizador visita ou até mesmo gravar as teclas do dispositivo [18].

3.2.1.1.4 Ransomware

Ransomware é um tipo de malware que encripta os dados do utilizador depois de executar um programa executável no dispositivo. Neste tipo de ataque, a chave de descriptografia é mantida até que o utilizador pague um resgate. É um dos tipos de *phishing* mais utilizados contra empresas ou organizações [23].

3.2.1.1.5 Rootkits

Um rootkit é um conjunto de programas, que tem como objetivo entrar na rede, e impedir que exista a deteção por parte dos sistemas de controlo, permitindo assim o controlo da segurança da rede por parte dos invasores [12].

3.2.1.2 Hosts File Poisoning

Quando o utilizador digita um endereço específico, o URL é traduzido num IP antes de aceder ao site, e o DNS é alterado, levando o utilizador a um site *phishing* [13].

3.2.1.3 Ataques de reconfiguração do sistema

Neste caso as configurações do computador são alteradas utilizando diferentes métodos, como por exemplo reconfiguração do sistema operativo ou modificação do DNS. Esta reconfiguração permite a monitorização do utilizador [13].

3.2.2 *Phishing* baseado em DNS (Pharming)

Inclui todas as formas de *phishing* que interferem com o DNS para que o utilizador seja redirecionado para o site malicioso, poluindo o cache DNS do utilizador com informações erradas. Embora o arquivo do host não faça parte do DNS, o envenenamento por arquivos do host é outra forma de *phishing* baseado em DNS. Por outro lado, comprometendo o DNS, os endereços IP genuínos serão modificados. O utilizador pode ser vítima de pharming mesmo clicando num link legítimo porque o DNS do site pode ser sequestrado por cibercriminosos [13].

3.2.3 *phishing* de injeção de conteúdo

phishing com injeção de conteúdo refere-se à inserção de conteúdo falso num site legítimo. O conteúdo pode ser colocado num site legítimo de três maneiras:

- através de uma vulnerabilidade de segurança e comprometimento de um servidor web;
- através de uma vulnerabilidade de Scripting cross-site (XSS) que é uma falha de programação que permite aos invasores inserir scripts do lado do cliente em páginas da Web, que serão visualizados pelos visitantes no site alvo;

- através de uma vulnerabilidade de injeção de Linguagem de Consulta Estruturada (SQL), que permite que hackers roubem informações do banco de dados do site, executando comandos de banco de dados num servidor remoto [26].

3.2.4 Man-in-the-middle *phishing*

O ataque Man In The Middle (MITM) é uma forma de *phishing*, no qual são inseridas comunicações entre duas partes (ou seja, o utilizador e o site legítimo) e tentam obter as informações de ambas as partes interceptando as comunicações da vítima. De tal forma que a mensagem é encaminhada para o invasor ao invés dos destinatários legítimos. O ataque do MITM ocorre através de várias técnicas, como "contaminação" pelo Address resolution protocol poisoning, falsificação de DNS, Trojan key loggers e URL obfuscation [20].

3.2.5 Mecanismo de pesquisa *phishing*

É criado um site malicioso com ofertas atraentes que utiliza técnicas de otimização de mecanismos de pesquisa para que seja indexado legitimamente, de modo que o site apareça quando o utilizador procura produtos ou serviços.

3.2.6 Ataques URL

Na maioria dos ataques de *phishing*, os phishers visam convencer um utilizador a clicar num determinado link que conecta a vítima a um servidor de *phishing* malicioso em vez do servidor de destino. Este tipo de ataque é realizado ofuscando o link real (URL) que o utilizador pretende conectar [20].

4 Medidas contra o *phishing*

Existem 3 estratégias base para combater o *phishing*:

- a consciencialização do utilizador de forma a que este seja capaz de identificar e reagir de forma adequada, quando confrontado com uma ameaça;
- utilização da lei como medida de dissuasão;
- soluções técnicas de deteção automática dos ataques em estágios iniciais.

Dentro das soluções técnicas para deteção automática de *phishing* existem essencialmente 3 categorias: Ativação baseada em endereços da web, ativação baseada em conteúdos/similaridade da página web e abordagem híbrida.

4.1 Ativação baseada em endereços da web

A URL é um endereço de rede constituído por protocolo, nome de domínio, extensão do domínio, caminho e nome do arquivo, a ativação baseada em endereços web, tem como finalidade a avaliação de cada uma destas estruturas. Os esquemas de avaliação podem ser realizados com diferentes metodologias: técnicas de deteção baseadas em listas, técnicas heurísticas de deteção baseadas em regras e técnicas de deteção baseadas em aprendizagem [8].

4.1.1 Técnicas de deteção baseadas em listas

Neste tipo de técnica existe uma lista com URLs que pode ser uma lista branca que contém URLs que são considerados fidedignos, ou uma lista negra com URLs que são considerados maliciosos. No caso de listas brancas, o acesso é fornecido apenas para os URLs que estão presentes na lista, enquanto na abordagem de lista negra o acesso é fornecido a qualquer URL diferente das que estão presentes na lista. A lista tem que ser constantemente atualizada [22].

4.1.2 Técnicas heurísticas de deteção baseadas em regras

As técnicas de heurísticas de avaliação de endereços web baseados em regras aplicam heurísticas com base nas normas de estrutura de um URL, para verificar a autenticidade da mesma [11].

4.1.3 Técnicas de deteção baseadas em aprendizagem

Algoritmos de aprendizagem como ML e deep learning são usados para detetar os ataques com base nos recursos extraídos da URL. Os recursos estatísticos e recursos de NLP dos URLs são extraídos e alimentados em algoritmos ML, como máquina vetorial de suporte (SVM), árvore de decisão, algoritmo ingénuo de Bayes, floresta aleatória etc. para classificação posterior. O classificador cria um modelo baseado na inferência extraída das amostras de treinamento. A URL suspeita é avaliada com base no modelo construído pelo classificador [25]

4.2 Ativação baseada em conteúdos/similaridade da página web

A falsificação de páginas web implica a cópia das fontes, layout, imagens e logótipos de forma que o site se pareça o mais possível ao original. Para detetar estes sites, é realizada a extração de vários recursos da página e avaliada a semelhança com o site original. Esta avaliação pode ser feita com base em esquemas de seleção baseada em conteúdos ou esquemas de deteção baseados em layout. No caso dos esquemas de deteção baseados em conteúdo, o conteúdo da página da Web serve como o principal parâmetro para classificar sites de *phishing*, São extraídas palavras-chave suspeitas, que são utilizadas como parâmetros de pesquisa para medir a acessibilidade da página web [30].

Nos esquemas de deteção baseados em similaridade, o layout do site é levado em consideração, podendo ser aplicada heurística ou algoritmos ML.

4.2.1 Cálculo de similaridade de páginas web baseado em regras heurísticas

No cálculo de similaridade de páginas web baseado em heurística, são extraídos da página suspeita, palavras-chave e recursos e são verificados na página da Web alvo utilizando métodos de pesquisa. Os recursos extraídos incluem recursos HTML, como número de links internos e externos, links vazios, formulário de login, comprimento de conteúdo HTML, janela de alarme, redirecionamento, informações ocultas/restritas, consistência entre marca de título e marca URL, consistência entre marca de link mais frequente e marca URL, recursos internos e externos, número da marca URL que aparece em HTML e assim por diante [15] e recursos CSS, como cor de propriedade, preenchimento em relação ao elemento no parágrafo, tamanho da fonte, borda, família de fontes e margens [16].

4.2.2 Avaliação de similaridade de páginas web baseada em algoritmos ML

Os recursos HTML, linguagem de marcação extensível, JavaScript (JS) e CSS são extraídos do código-fonte da página web e são alimentados em algoritmos ML para classificação [30].

4.3 Abordagem híbrida

As abordagens híbridas para deteção de *phishing* na web são uma combinação dos esquemas de deteção de ativação baseada em endereços da web e ativação baseada em conteúdos/similaridade da página web [9].

5 Exemplos de aplicações/ferramentas Anti-*phishing*

Tendo como base as ferramentas apresentadas na secção 2, vamos aprofundar algumas metodologias diferentes. Primeiro mostramos a PhishBlock, que apresenta uma abordagem híbrida, e de seguida apresentamos 2 metodologias mais recentes que utilizam inteligência artificial para a deteção de sites *phishing*.

5.1 PhishBlock

Fahmy Ghoneim (2011) desenvolveram o PhishBlock que é uma ferramenta para deteção dinâmica e proativa de sites falsificados, que junta sistemas de pesquisa e classificação num programa independente do navegador, foi desenvolvido o PhishBlock que usa um código-fonte aberto.
[7].

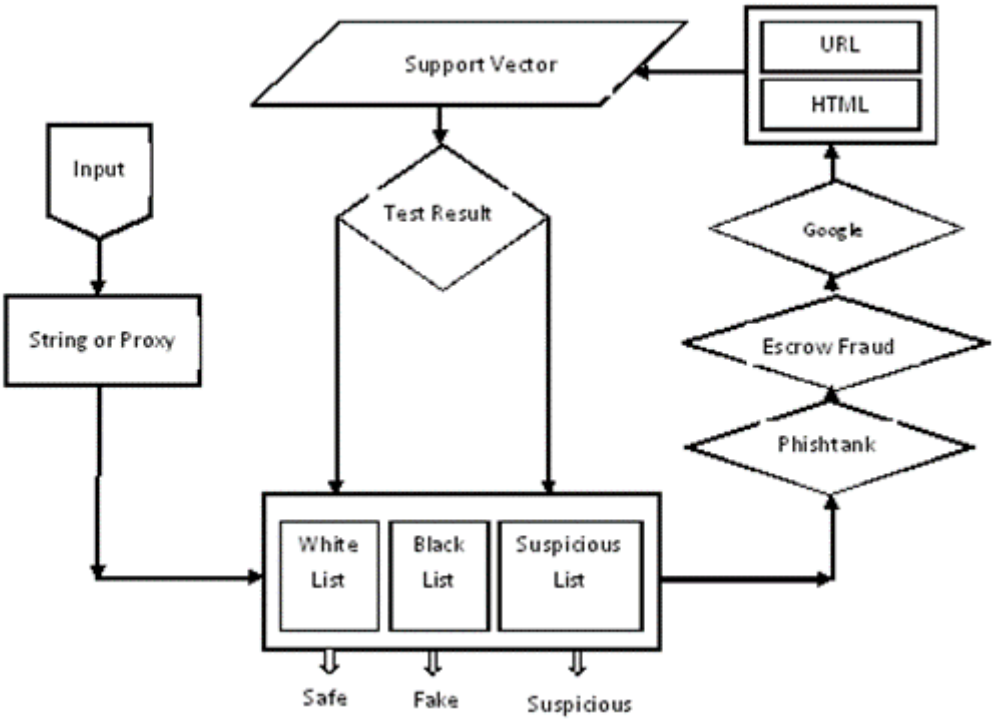

Figura 1: Diagrama do PhishBlock

A figura 1, apresenta o diagrama do PhishBlock, onde é possível perceber como funciona a ferramenta desenvolvida pelos autores. No PhishBlock, são utilizadas três listas locais (lista preta, branca e suspeita) por onde a URL a ser testada passa. Se for encontrada em qualquer um deles, é apresentada uma mensagem ao utilizador (Falso, Seguro ou Suspeito), caso contrário, a URL é passa para os servidores globais para ser testada pelo SVM.

No PhishBlock, são utilizados três servidores (Phishtank, Google e Escrow Fraud). A URL a ser testada passa pelo Phishtank, onde é detectada como um site falso ou desconhecido, se for considerado desconhecido passa para o Escrow Fraud, caso contrário é adicionada à lista negra do PhishBlock. O mesmo procedimento ocorre no Escrow Fraud, onde a URL passa por o Google se não for desconhecida, caso contrário, ela é adicionada à lista negra do PhishBlock.

O sistema de classificação PhishBlock é implementado utilizando redes neurais. As redes neurais, são utilizadas para extrair padrões e detectar tendências que são muito complexas. A rede neuronal utilizada é a máquina vetorial de suporte (SVM). Os SVMs são um conjunto de métodos de aprendizagem supervisionados relacionados, utilizados para classificação. Dado um conjunto de exemplos de treino, cada um marcado como pertencente a uma das duas categorias, um algoritmo de treino SVM constrói um modelo que prevê se um novo exemplo se encaixa numa categoria ou noutra.

A ferramenta de anti-*phishing* PhishBlock apresentou uma accuracy de 95% e uma taxa de falsos positivos muito baixa (0,1%). Este estudo sugere que sistemas que dependem apenas de mecanismos de pesquisa ou sistemas de classificação que utilizam um pequeno conjunto de recursos são ineficazes no combate ao *phishing*.

5.2 Deteção de *phishing* baseada no AV-BMEO com KNN

Esta abordagem foi desenvolvida por Minocha Singh (2022) e utiliza o AV-BMEO que é uma versão binária do Modified Equilibrium Optimizer (MEO) que utiliza a função de transferência AV e a aprendizagem baseada em oposição para melhorar a capacidade de exploração do MEO binário [17].

As funções de transferência são utilizadas para converter as soluções de domínio contínuo em soluções de domínio binário na seleção de recursos. A utilização das funções de transferência melhora o desempenho de classificação aumentando a capacidade de exploração do algoritmo.

A seleção dos recursos é um problema binário que seleciona um recurso específico com base na sua contribuição para a classificação. O espaço de pesquisa é um hipercubo com soluções athwart. O que reduz o problema a um array binário n-dimensional.

O otimizador BMEO é utilizado para a seleção de recursos e otimização dos hiperparâmetros, O classificador kNN para o valor otimizado com k recursos selecionados é utilizado para a classificação.

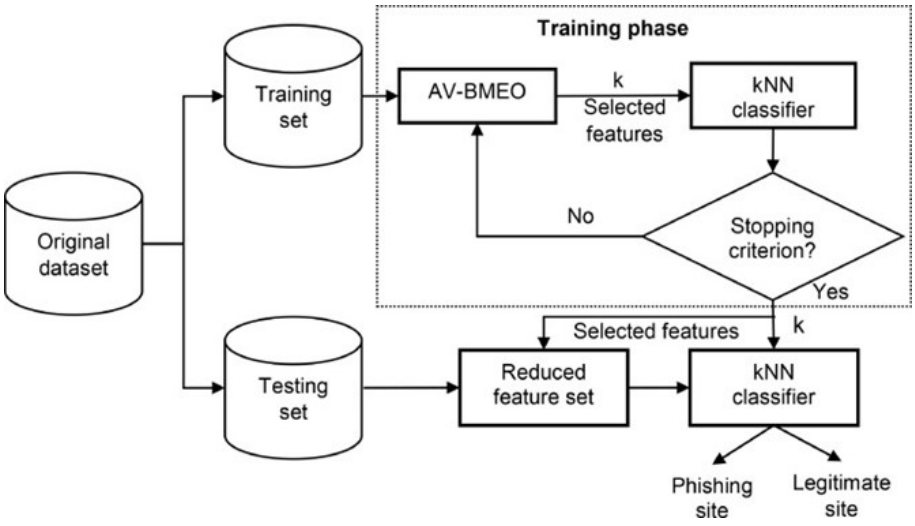

Figura 2: Sistema de deteção de *phishing* proposto pelos autores

A figura 2 apresenta o esquema do sistema desenvolvido. O modelo utiliza com conjunto de dados de entrada que contém recursos extraídos de sites *phishing* e de sites legítimos. Os recursos contêm domínio, URL, HTML e Java Script. Estes dados foram divididos em dados de treino e teste. O algoritmo AV-BMEO realiza a seleção do recurso do wrapper e a otimização do hiperparâmetro k do classificador kNN. kNN é um classificador que armazena os dados na fase de treino. Ele atribui uma classe a novos dados durante a fase de testes com base na classe de vizinhos mais próximos para um novo conjunto de dados. O AV-BMEO com o classificador kNN realiza a otimização até que o critério de paragem seja satisfeito, nesse momento a fase de treino é concluída e são retornadas as características selecionadas e o hiperparâmetro k. Em seguida o classificador kNN realiza a classificação nos dados teste com base nos recursos selecionados anteriormente. Sendo assim possível obter uma classificação de um site como legítimo ou não legítimo (*phishing*).

Os autores efetuaram vários testes e comparações com outros algoritmos semelhante. Obtendo um valor de 96.36 e 97.46% de accuracy (precisão) para 2 conjuntos de dados diferentes.

5.3 Deteção de *phishing* através de um modelo de deep learning

Esta abordagem foi desenvolvida por Alshehri et al. (2022) e utiliza métodos de aprendizagem profunda, nomeadamente um modelo de deep learning para deteção de sites *phishing*. O modelo de rede utilizada é o deep neural network (DNN) [3].

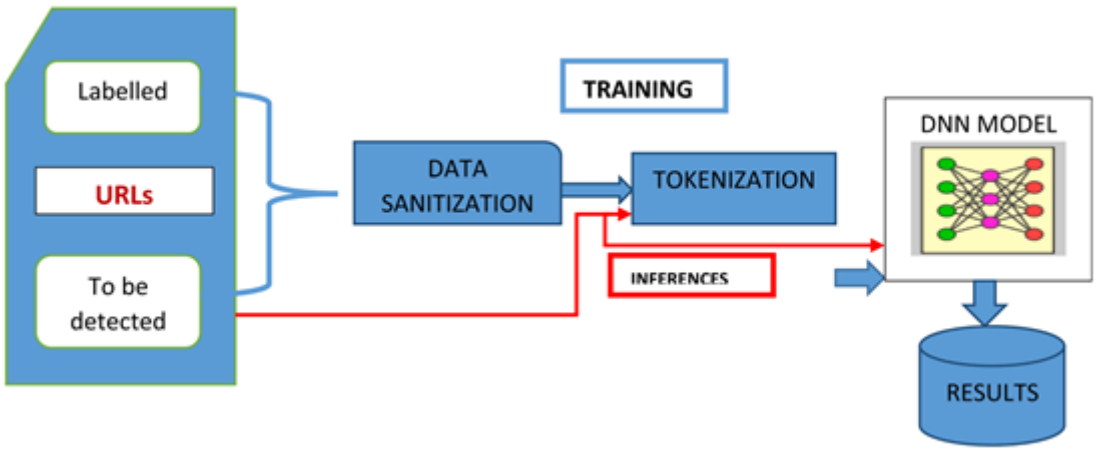

Figura 3: Metodologia do modelo proposto pelos autores [3]

A figura 3 apresenta a metodologia geral utilizada pelos autores onde a etapa inicial do modelo é a sanitização de dados, na qual os prefixos comuns da URL, (HTTP:// e HTTPS://), são removidos para aumentar o desempenho da deteção de sites *phishing*, este procedimento permite reduzir o impacto das representações URL nos diferentes conjuntos de dados. Posteriormente ocorre a tokenização que é usada para vetorizar todos os caracteres que estão presentes na URL. O método utiliza um conceito de tokenização de caracteres para evitar usar o significado das palavras na URL.

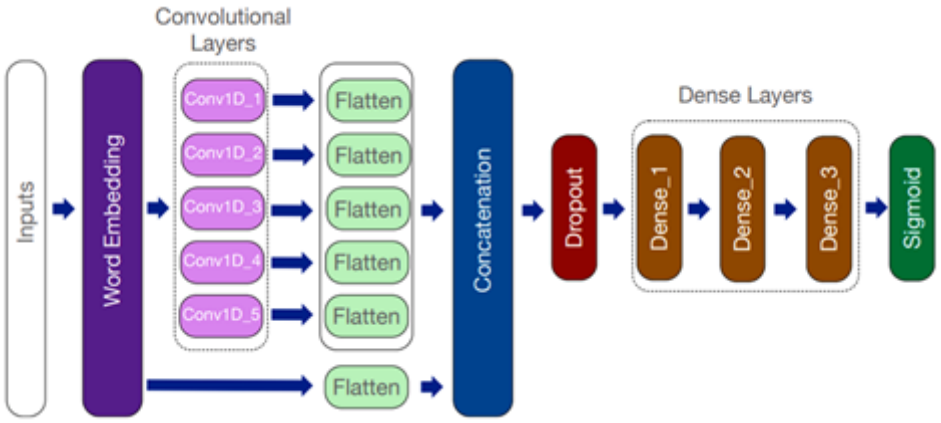

Figura 4: Rede utilizada pelos autores [3]

A figura 4 apresenta a rede neuronal desenvolvida e utilizada pelos autores. A camada embedded é normalmente usada como o primeiro nível do modelo DNN para NLP. Esta camada produz um vetor. A palavra embedding dos coeficientes de cada vetor é posteriormente retornada da camada embedded, podendo representar a relação entre os diferentes caracteres. Este procedimento ajuda a melhorar o desempenho da NLP.

São utilizadas quatro camadas convolucionais. Em cada uma das camadas é aplicado um kernel e um filtro, para extrair os recursos importantes e remover informações menos interessantes. A multiplicação é realizada com base nos elementos e ocorre um resumo da operação entre os filtros e a parte apropriada dos dados. São utilizadas camadas paralelas. Cada camada é projetada para considerar uma única janela de caracteres consecutivos e extrair os recursos.

A camada de concatenação foi projetada para concatenar os recursos das camadas anteriores de forma a permitir o seu processamento. Ao contrário da concatenação normal, neste caso, as saídas das camadas convolucionais são combinadas com a camada embedded. Isto preserva o contexto original das informações que são usadas posteriormente para a detecção de URLs de *phishing*.

A camada de dropout tem como objetivo prevenir o over-fiting durante o treino. Os neurónios são selecionados de forma aleatória e ignorados durante o treino, sendo removidos temporariamente e os seus pesos são impedidos de atualizar na época seguinte.

As camadas densas são totalmente concatenadas, conferindo ao modelo uma capacidade de extração das características mais relevantes. O principal objetivo desta camada é a análise dos padrões originados nas camadas convolucionais. A camada de saída é uma função sigmoide, pois esta admite valores entre 0 e 1, sendo assim possível obter um valor de probabilidade no fim do modelo.

Foram utilizadas 10.234 URLs no total, sendo 7.234 URLs de sites legítimos e 3.004 de *phishing*. Os dados foram divididos em dados de treino e teste na proporção de 80:20. Primeiro, foram utilizadas as URLs rotuladas para treinar o modelo DNN usando dispositivos computacionais, como racks de servidores. Esses modelos de DNN treinados foram posteriormente transferidos para o sensor, que atuou como um detector de *phishing*. Sempre que o sensor recebia uma solicitação de URL, ele realizava o processo de deteção usando o modelo DNN integrado antes que o DNS fosse adquirido. Quando o URL foi determinado como malicioso, um alarme foi iniciado pelo sensor. As instâncias TP da deteção foram monitoradas e gravadas para testar a precisão do modelo. O modelo proposto obteve uma accuracy de 98,13% .

6 Conclusão

O *phishing* tem vindo a aumentar nos últimos anos, o que leva a um impacto bastante negativo tanto a nível económico como a nível social. Cada vez mais empresas têm vindo a ser vítimas de ataques,

aumentando assim a preocupação, bem como a urgência em encontrar métodos eficazes que impeçam ataques. Várias ferramentas têm sido desenvolvidas com o objetivo de diminuir o número de ataque e a suscetibilidade das pessoas aos mesmos. Inicialmente as técnicas de combate ao *phishing* tinham por base o desenvolvimento de extensões para o browser que funcionavam com listas e métodos tradicionais de pesquisa, posteriormente foram desenvolvidos métodos mais complexos e que se mostraram mais eficazes. Atualmente a maioria dos métodos desenvolvidos são da área de inteligência artificial, como machine learning e deep learning, que permite uma atualização constante do modelo, sem interação humana. No entanto, ainda temos um, longo caminho a percorrer, pois os métodos de ataque estão em contante mudança sendo cada vez mais sofisticados e difíceis de detetar.

Referências

[1] Maher Aburrous, M. A. Hossain, Fadi Thabatah, and Keshav Dahal. Intelligent phishing website detection system using fuzzy techniques. In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, pages 1–6, 2008. **https://doi.org/10.1109/ICTTA.2008.4530019**.

[2] Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3, 2021. **https://doi.org/10.3389/fcomp.2021.563060**.

[3] Mohammed Alshehri, Ahed Abugabah, Abdullah Algarni, and Sultan Almotairi. Character-level word encoding deep learning model for combating cyber threats in phishing url detection. *Computers and Electrical Engineering*, 100:107868, 2022. **https://doi.org/10.1016/j.compeleceng.2022.107868**.

[4] APAV. Folha informativa phishing, 2013.

[5] Abdullateef O Balogun, Kayode S Adewole, Muiz O Raheem, Oluwatobi N Akande, Fatima E Usman-Hamza, Modinat A Mabayoje, Abimbola G Akintola, Ayisat W Asaju-Gbolagade, Muhammed K Jimoh, Rasheed G Jimoh, et al. Improving the phishing website detection using empirical analysis of function tree and its variants. *Heliyon*, 7(7):e07437, 2021. **https://doi.org/10.1016/j.heliyon.2021.e07437**.

[6] Neil Chou, Robert Ledesma, Teraguchi Yuka, and John C Mitchell. Client-side defense against web-based identity theft. *Computer Science Department, Stanford University. Available: **http://crypto. stanford. edu/SpoofGuard/webspoof. pdf***, 2004.

[7] Hossam M.A. Fahmy and Salma A. Ghoneim. Phishblock: A hybrid anti-phishing tool. In *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, pages 1–5, 2011. **https://doi.org/10.1109/CCCA.2011.6031523**.

[8] Varshney Gaurav, Misra Manoj, and K Atrey Pradeep. A survey and classification of web phishing detection schemes. *Security and Communication Networks*, 9(18):6266–6284, 2016. **https://doi.org/10.1002/sec.1674**.

[9] R. Gowtham and Ilango Krishnamurthi. A comprehensive and efficacious architecture for detecting phishing webpages. *Computers Security*, 40:23–37, 2014. **https://doi.org/10.1016/j.cose.2013.10.004**.

[10] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, oct 2007. **https://doi.org/10.1145/1290958.1290968**.

[11] S. Carolin Jeeva and Elijah Blessing Rajsingh. Intelligent phishing url detection using association rule mining. *Human-centric Computing and Information Sciences*, 6(1):10, Jul 2016. **https://doi.org/10.1186/s13673-016-0064-3**.

[12] Robert Karamagi. A review of factors affecting the effectiveness of phishing. *Computer and Information Science*, 15(1), 2022. **https://doi.org/10.5539/cis.v15n1p20**.

[13] Latika Kharb. What is pharming? *January* 2017.

[14] Engin Kirda and Christopher Kruegel. Protecting Users against Phishing Attacks. *The Computer Journal*, 49(5):554–561, 01 2006. **https://doi.org/10.1093/comjnl/bxh169**.

[15] Yukun Li, Zhenguo Yang, Xu Chen, Huaping Yuan, and Wenyin Liu. A stacking model using url and html features for phishing webpage detection. *Future Generation Computer Systems*, 94:27–39, 2019. **https://doi.org/10.1016/j.future.2018.11.004**.

[16] Jian Mao, Wenqian Tian, Pei Li, Tao Wei, and Zhenkai Liang. Phishing-alarm: Robust and efficient phishing detection via page component similarity. *IEEE Access*, 5:17020–17030, 2017. **https://doi.org/10.1109/ACCESS.2017.2743528**.

[17] Sachin Minocha and Birmohan Singh. A novel phishing detection system using binary modified equilibrium optimizer for feature selection. *Computers Electrical Engineering*, 98:107689, 2022. **https://doi.org/10.1016/j.compeleceng.2022.107689**.

[18] Latto N. O que é adware e como evitá-lo? *Avast Academy*, 2021. **https://www.avast.com/pt-br/c-adware**.

[19] DaeHun Nyang, Aziz Mohaisen, and Jeonil Kang. Keylogging-resistant visual authentication protocols. *IEEE Transactions on Mobile Computing*, 13(11):2566–2579, 2014. **https://doi.org/10.1109/TMC.2014.2307331**.

[20] Gunter Ollmann. The phishing guide understanding & preventing phishing attacks. *NGS Software Insight Security Research*, 2004.

[21] Seguin P. *Avast Academy*, 2022. **https://www.avast.com/pt-br/c-spyware**.

[22] Issa Qabajeh, Fadi Thabtah, and Francisco Chiclana. A recent review of conventional vs. automated cybersecurity anti-phishing techniques. *Computer Science Review*, 29:44–55, 2018. **https://doi.org/10.1016/j.cosrev.2018.05.003**.

[23] T.R. Reshmi. Information security breaches due to ransomware attacks - a systematic literature review. *International Journal of Information Management Data Insights*, 1(2):100013, 2021. **https://doi.org/10.1016/j.jjimei.2021.100013**.

[24] Troy Ronda, Stefan Saroiu, and Alec Wolman. Itrustpage: A user-assisted anti-phishing tool. In *Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008*, Eurosys '08, page 261–272, New York, NY, USA, 2008. Association for Computing Machinery. **https://doi.org/10.1145/1352592.1352620**.

[25] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, and Banu Diri. Machine learning based phishing detection from urls. *Expert Systems with Applications*, 117:345–357, 2019. **https://doi.org/10.1016/j.eswa.2018.09.029**.

[26] Marcelo Invert Salas Palma, Paulo Licio De Geus, and Eliane Martins. Security testing methodology for evaluation of web services robustness - case: Xml injection. In *2015 IEEE World Congress on Services*, pages 303–310, 2015. **https://doi.org/10.1109/SERVICES.2015.53**.

[27] Hossain Shahriar and Vamshee Krishna Devendran. Classification of clickjacking attacks and detection techniques. *Information Security Journal: A Global Perspective*, 23(4-6):137–147, 2014. **https://doi.org/10.1080/19393555.2014.931489** .

[28] Xinyuan Wang, Ruishan Zhang, Xiaohui Yang, Xuxian Jiang, and Duminda Wijesekera. Voice pharming attack and the trust of voip. SecureComm '08, New York, NY, USA, 2008. Association for Computing Machinery. **https://doi.org/10.1145/1460877.1460908**.

[29] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Trans. Inf. Syst. Secur.*, 14(2), sep 2011. **https://doi.org/10.1145/2019599.2019606**.

[30] Yue Zhang, Jason I. Hong, and Lorrie F. Cranor. Cantina: A content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 639–648, New York, NY, USA, 2007. Association for Computing Machinery. **https://doi.org/10.1145/1242572.1242659**.