DIAHPhish: Uma solução baseada em Redes Neurais Siamesas para detecção de ataques homográficos em páginas phishing direcionadas

Lucas C. Teixeira [1], **Bruno J. T. Fernandes**[1], **Carlo M. R. Silva**[1], **Julio C. G. Barros**[1]

[1]Escola Politécnica (POLI) – Universidade de Pernambuco (UPE)

{lct, bjtf, cmrs, jcgb}@ecomp.poli.br

Abstract. *Phishing is one of the most popular mechanisms for applying virtual scams in activity. Much of the effectiveness of phishing attacks lies in their ability to trick the user into convincing them that they are accessing a genuine service. For such a function, a significant portion of the attacks explore the application of homographic terms to check the reliability of the attack. In this scenario, the study proposes an autonomous approach, based on an LSTM recurrent Siamese neural network, capable of identifying the presence of homographic terms in parts of the URL and content of phishing pages. As a result, the proposed model proved to be highly efficient in detecting malicious terms, reaching an average assertiveness rate of more than 99.50%.*

Resumo. *O phishing é um dos mecanismos mais populares para aplicar golpes virtuais em atividades. Grande parte da eficácia dos ataques de phishing reside na capacidade de enganar o usuário, convencendo-o de que está acessando um serviço legítimo. Para essa função, uma parte significativa dos ataques explora a aplicação de termos homográficos para verificar a confiabilidade do ataque. Nesse cenário, o estudo propõe uma abordagem autônoma, baseada em uma rede neural siamesa recorrente LSTM, capaz de identificar a presença de termos homográficos em partes da URL e conteúdo de páginas de phishing. Como resultado, o modelo proposto mostrou-se altamente eficiente na detecção de termos maliciosos, alcançando uma taxa de assertividade de mais de 99,50%.*

1. Introdução

Phishing é um dos golpes cibernéticos mais populares do mundo e tem o Brasil como seu principal alvo[1]. Responsável por mais de 50% dos ataques cibernéticos a cartões de crédito[2], *phishing* é caracterizado pela utilização de páginas falsas, que se assemelham a populares sites da web, buscando construir um ambiente confiável para que seus usuários forneçam dados sigilosos, a exemplo de senhas ou dados de cartões de crédito.

Buscando conferir fidedignidade aos ataques, muitas dessas fraudes exploram a suscetibilidade dos usuários em não identificar a presença de ataques homográficos, recurso caracterizado pela aplicação de termos com grafias semelhantes, mas que não se referem à página esperada. Um exemplo dessa exploração é o registro de domínios com pequenas variações gramaticais, produzidos por meio de engenharias astutas, levando o

[1]http://bit.ly/389cHRl
[2]https://bit.ly/2mv2uvN

usuário a acreditar que está no ambiente seguro da marca desejada [Piredda et al. 2017]. Dessa forma, a fraude explora a suscetibilidade do usuário através da confiabilidade, conferida pelo termo homográfico[3].

Nessa circunstância, várias soluções para deter os ataques homográficos têm sido propostas [Chiba et al. 2018] [Le Pochat et al. 2019], porém, existem desafios a serem superados. Cita-se, como exemplo, o esforço para construir e controlar o conjunto de termos utilizados para conferir confiabilidade ao ataque, quando novos termos aparecem com alta frequência, permitindo sua exploração por novos phishings. Outro entrave é o alto volume de termos com pequenas variações gramaticais, compostos por meio de engenharias que exploram comportamentos como inserção, remoção, substituição de letras, entre outros. Dessa forma, surgem diversas variações que resultam em falsos positivos e negativos, dificultando a construção de um padrão textual.

Diante desse cenário, o trabalho propõe DIAHPhish, uma metodologia inteligente baseada em redes neurais siamesas, capaz de identificar de forma autônoma e precisa a ocorrência de ataques homográficos em páginas *phishing*. Como diferencial, o modelo tem por objetivo a identificação da marca-alvo do ataque, através de dados textuais extraídos das URLs maliciosas, sem o uso de volumosas bases de dados *phishing*, com reduzido impacto financeiro e alta precisão em sua detecção.

2. Contextualização

Nesta seção, será apresentado o conceito e a caracterização de ataques homográficos, juntamente a uma breve contextualização de outras pesquisas dentro da temática do estudo.

2.1. Ataques homográficos

Comumente aplicado a partes da URL, como domínios e subdomínios, ataques homográficos reúnem um conjunto de técnicas que visam manipular, a partir de engenharias enganosas, caracteres textuais exibidos ao usuário [Spaulding et al. 2017]. Com a finalidade de conferir fidedignidade ao ataque, explorando distração ou ímpeto, ataques homógrafos objetivam extrair dados sensíveis, como número de cartões de crédito ou chaves de acesso, podendo assim ser caracterizado como *phishing* [Tahir et al. 2018].

De modo geral, a exploração de ataques homográficos se dá através de *Cybersquatting* e *Typosquatting*. A primeira técnica tem foco na exploração de termos genuinamente corretos que remetem à marca-alvo, como sinônimos, palavras-chave, abreviaturas ou jargões, termos denominados, por este trabalho, como palavras em *plain-text*.

Definido pelo registro de domínios que fazem menção textual a outra determinada marca [Buber et al. 2017], *cybersquatting* não é considerado um crime virtual. No entanto, sua utilização abre caminho para aplicação de diversos golpes cibernéticos, a exemplo de *phishing*, que exploram o perfil promissor da aplicação da técnica como um mecanismo de enriquecimento textual dos golpes. Um exemplo dessa aplicação é no endereço "http://lotericas.caixa.apostas-online.com.br" que sugere um serviço ligado às Loterias Caixa, sob responsabilidade da Caixa Econômica Federal.

A técnica seguinte é, assim como a primeira, baseada na manipulação de caracteres textuais aplicados a partes da página fraudulenta. *Typosquatting* é a ação de regis-

[3]Entende-se termos homográfico como: palavra com alteração de sentido ou composição gramatical.

trar domínios com erros gramaticais deliberados objetivando encaminhar o usuário para uma página fraudulenta, através da fidedignidade conferida pelo domínio, a exemplo de "go0gle.com", "facbook.com" e "netfliix.com" [Spaulding et al. 2017].

A primeira menção de sua aplicação foi apresentada em 1998, através de uma publicação realizada pelo The New York Law Journal [Gilwit 2003], mas o primeiro estudo em larga escala só foi desenvolvido em 2003, onde foram identificados 8.800 domínios dessa natureza que, em quase toda sua totalidade, estão vinculados a um mesmo indivíduo, John Zuccarini, que utilizava essa arquitetura para redirecionar os usuários para páginas de conteúdo adulto, onde eram expostos a diversos *softwares* maliciosos [Spaulding et al. 2016], comportamento que caracteriza a ação como *phishing*.

2.2. Trabalhos relacionados

Estudos que abordam ataques homográficos e páginas *phishing* podem ser observados em periódicos e conferências de alto impacto acadêmico. Dentre esses, grande parte dos trabalhos concentram seus esforços na análise dos comportamentos empregados a termos *Typosquatting*. Nesse cerne, podemos destacar o trabalho desenvolvido por Tobias et. al. [Dam et al. 2019], que observou 7.176 páginas *web*, coletadas da lista Alexa Top 1 Million, classificadas como *"typosquatting"*, e que apresentavam *pop-ups* em sua composição. Após a seleção, as páginas foram varridas utilizando uma adaptação do *framework MiningHunter*, que foi originalmente desenvolvido para identificar campanhas de mineração de criptomoedas.

Não obstante, Quinkert et. al. [Quinkert et al. 2019] fazem uma análise longitudinal quanto aos comportamentos homográficos presentes em páginas web fraudulentas direcionadas a empresas de tecnologia e sistemas bancários. Os autores desenvolvem uma metodologia em dois processos, com início na extração de domínios através do "Domainlists.io", e logo em seguida, os registros são avaliados logitudinalmente, buscando identificar a presença de menções homográficas às marcas participantes.

Já os trabalhos voltados para detecção de ataques homográficos podem ser divididos em dois grupos. O primeiro está dedicado a identificar os mais prováveis erros gramaticais propositais empregados à marca, abrindo espaço para as empresas realizarem o registro preventivo destes domínios. Nesse sentido, o trabalho desenvolvido por Ahmad, Parvez e Iqbal [Ahmad et al. 2019] propõe o modelo *TypoWriter*, uma aplicação baseada em redes neurais recorrentes (RNN) para previsão de termos homográficos voltados a uma determinada marca-alvo. Com essa arquiterura, o modelo é capaz de fornecer uma curta relação de termos homográficos, que possuem uma maior probabilidade de aplicação a uma marca em questão, possibilitando os registros defensivos dos domínios.

O segundo grupo de propostas para identificação da presença de ataques homográficos está voltado à identificação da marca-alvo a partir da similaridade entre o termo homográfico e o fidedigno. Com esse objetivo, Lio et. al. [Liu et al. 2016] propõem *TypoPegging*, um modelo baseado em Distância de Levenshtein para a identificação de *typosquatting* em páginas *phishing*. Com uma métrica capaz de aferir a diferença caractere-a-caractere de entradas textuais, o modelo se propõe a realizar a autenticação de domínios suspeitos com base em sua semelhança visual a domínios comprovadamente genuínos.

No mesmo sentido, Moubayed et al. [Moubayed et al. 2018] propõem um modelo híbrido para classificação de domínios *phishing*, formado pelos algoritmos: *decision trees*

(C4.5), *K-nearest neighbors* (K-NN), *logistic regression* (LR), *Naive Bayesian* (NB), e *Support Vector Machines* (SVM). Unidos através de voto majoritário, a arquitetura analisa oito aspectos textuais elencados através de um mapeamento estatístico, aplicado a um conjunto de URLs maliciosas, para construir a base de treinamento dos classificadores.

Não obstante, Ya et al. [Ya et al. 2018] apresentam *TypoEval*, que assim como o modelo proposto neste estudo, é baseado em redes neurais siamesas recorrentes, composta por duas redes LSTM, agrupadas por uma função de similaridade euclidiana. O modelo obteve ótimos resultados ao ser treinado e testado em uma base de dados sintéticos, elaborada pela ferramenta *Typofinder*. Ao fim do experimento, pode-se concluir que *TypoEval* possui uma arquitetura eficiente para detecção de domínios *typosquatting*, chegando a atingir a marca de 97,19% de precisão, na identificação *typosquatting*.

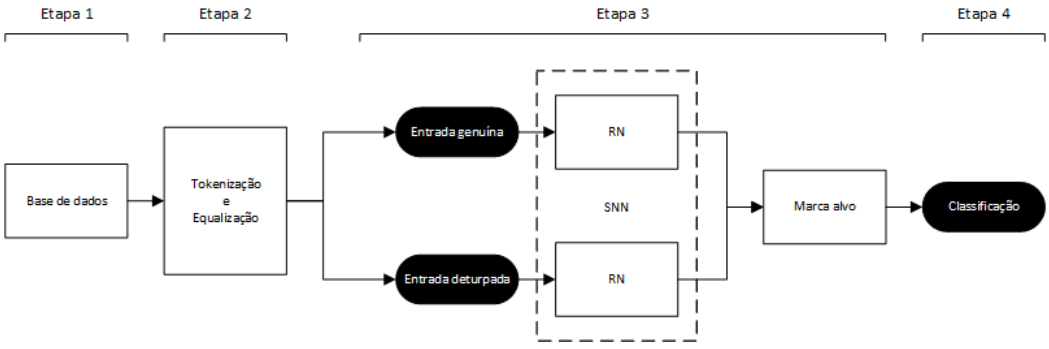
Na pesquisa de Zhu et. al. [Zhu et al. 2018], os autores apresentam uma solução para detecção de *phishing* baseada na análise de vários *embeddings* de palavras com diferentes algoritmos de aprendizado de máquina, criando um conjunto de modelos a partir desses dicionários. A técnica desenvolvida caracteriza-se pela independência de informações externas ao site verificado, como características de classificação de página ou resultados de mecanismos de busca.

Diante o exposto, é possível observar a presença de diversas fragilidades nas propostas, a exemplo: alto impacto financeiro e o total direcionamento dos modelos à identificação de ataques *typosquatting*. Neste cenário, DIAHPhish destaca-se ao proporcionar uma solução de baixo impacto financeiro, sensível a *typosquatting* e *cybersquatting* e que permite sua aplicabilidade a quaisquer partes textuais das páginas maliciosas.

3. DIAHPhish

O modelo de identificação de ataques homográficos em páginas *phishing* terá a função de identificar a utilização de marcas ou termos que remetem às mesmas (*typosquatting* ou *cybersquatting*) em conteúdos textuais aplicados às páginas maliciosas. Para tal função, a arquitetura utilizará uma metodologia baseada em redes neurais siamesas, onde os termos em linguagem corrente serão convertidos em sequências numéricas e comparados a partir de uma métrica de similaridade. A arquitetura proposta está dividida em quatro etapas, como apresentado na Figura 1.

Figura 1. Etapas da metodologia.



A primeira etapa da metodologia é a construção de uma base de dados. Após uma busca minuciosa na literatura, não foram identificadas bases rotuladas com características

suficientes acerca de ataques homográficos em páginas *phishing*, sejam esses através de *typosquatting* ou *cybersquatting*. Sendo assim, fez-se necessário a construção de um protótipo para geração de termos deturpados (*typosquatting*) sintéticos para execução do experimento, uma vez que a construção de uma base com ataques genuínos exige um período de tempo que não obedece às restrições temporais deste estudo.

Dessa forma, foram selecionadas 30 marcas-alvo, extraídas a partir de uma base de dados composta por 57.356 registros da popular plataforma *PhishTank*. Uma vez selecionados, os rótulos foram aplicados exaustivamente, de forma aleatória, ao conjunto de características comuns a termos homográficos evidenciado em um estudo anterior[Teixeira et al. 2021]. Estas são:

- **#01. Inserir letra de tecla vizinha no teclado**: o termo do google para goopgle.
- **#02. Inserir pluralidade**: por exemplo, o termo do google para googles.
- **#03. Inserir letra de forma repetida**: o termo do google para gooogle.
- **#04. Inserir separadores**: por exemplo, o termo "googledrive" por "google-drive" ou "google_drive" ou "google.drive".
- **#05. Omitir letra**: por exemplo, o termo google.com por "goole" ou "go-ogl.com".
- **#06. Trocar por tecla vizinha no teclado**: o termo google por giogle.
- **#07. Trocar posição de uma letra**: por exemplo, o termo google para ogogle.
- **#08. Trocar caractere por semelhança**: por exemplo, o termo do Google para goog1e ou go0gle (substituindo a letra o por, respectivamente, um número 0).
- **#09. Trocar vogal**: ex. o termo google para gaagle.
- **#10. Simulação de TLD**: por exemplo, o termo google.com para google-com.tk.

É relevante ressaltar que essa lista é originalmente composta por treze características, porém, os comportamentos "Sequestro de TLD", "Mudança de letra que mantem a mesma fonética" e "Omissão de SLD", foram desconsiderados , pois são de baixa aplicação, alto esforço para replicação ou de uso exclusivo em partes da URL [Teixeira et al. 2021].

Ao fim das manipulações, obtivemos uma base de dados rotulada pela marca-alvo, composta por 33.077 termos distorcidos (*typosquatting*) e termos genuínos (*cybersquatting*), que em seguida foram divididos em três conjuntos, respeitando as proporções de 60% para treino, 20% para teste e 20% para validação. Outro ponto a ressaltar é que, para este estudo, adotou-se apenas o nome da marca-alvo como termo *cybersquatting* como, por exemplo, para as marcas-alvo "Banco do Brasil" que contou com o *cybersquatting* "bancodobrasil". Essa escolha se deu pelo alto custo na definição de uma metodologia para extração de outros termos precedidos de grafia correta que remetem às marcas-alvo[4].

A próxima etapa é a execução do pré-processamento da base. Aqui, os dados gerados na etapa anterior são manipulados em dois processos: (I) tokenização e equalização; (II) emparelhamento. As redes neurais não são capazes de lidar com dados complexos, como palavras ou caracteres, sendo necessário converter dados alfabéticos em rótulos numéricos, processo denominado de tokenização. O processo de conversão se inicia com a construção de um vocabulário, onde caracteres ou palavras são vinculados a uma chave numérica (token), que servirá como identificador do termo.

[4]Mais detalhes quanto a construção e acesso a base de dados estão disponiveis em: https://github.com/LucasCTeixeira/DIAHPhish_Dataset.git

Dessa forma, os termos deixam de ser representados por uma sequência de caracteres textuais e passam a ser representados por uma sequência numérica, como por exemplo no termo “google” que, após a tokenização, é representado pela sequência “[1, 2, 2 , 1, 3, 4]”. O dicionário aplicado ao experimento será apresentado na Seção 5.1.

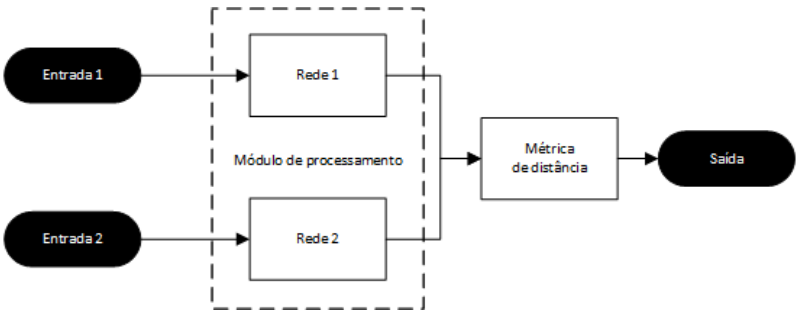
Para além das limitações no trato de dados textuais, as redes neurais são incapazes de atuar sobre dados com tamanhos variados, que é o caso dos termos recém tokenizados. Uma simples solução para o problema é a inclusão de valores nulos de forma que as entradas tokenizadas sejam equalizadas para um tamanho fixo, ou seja, entradas como “[1, 2 , 2, 1, 3, 4]” e “[1, 2, 3]”, podem ser representados como, por exemplo, uma sequência de comprimento 8: “[0, 0, 1, 2, 2, 1, 3 , 4]” e “[0, 0, 0, 0, 0, 1, 2, 3]”. Nesse exemplo, o valor nulo é representado pelo numeral zero, que foi adicionado à esquerda dos termos tokenizados, de forma que ambas as sequências possuam a quantidade de caracteres pré-estabelecida.

Já no pré-processamento e emparelhamento das entradas, os dados são agrupados aleatoriamente em pares, que recebem um rótulo com base em suas classes originárias. Sendo assim, para este estudo, adotou-se o valor “1”, como rótulo para representar pares da mesma classe, e “0” para representar pares pertencentes a classes diferentes.

A terceira etapa é a identificação da marca-alvo do ataque através do modelo baseado em aprendizado por representação, (Rede Neural Siamesa) uma arquitetura capaz de aprender representações discriminativas, através da extração de características com padrões sutis, e compará-los através de uma métrica de similaridade. Para tanto, o modelo recebe como entrada pares de textos, já codificados como conjuntos numéricos, e tem como objetivo determinar se os pares pertencem à mesma classe (marca).

A rede neural siamesa é composta por uma dupla de redes neurais artificiais, responsáveis pela identificação de características inerentes a cada entrada, e por uma única camada densa de saída, responsável por concentrar as características extraídas em um conjunto de dados de tamanho pré-definido. Ao fim desse processo, as saídas são submetidas a uma função de similaridade que deverá determinar a equidade ou não das entradas, uma vez que entradas de uma mesma marca deverão possuir distâncias mais curtas, enquanto pares de marcas distintas possuam distâncias mais longas. O fluxo do modelo de detecção de marcas pode ser observado na Figura 2.

Figura 2. Fluxo de uma Rede Neural Siamesa.



O modelo base de uma rede neural siamesa é escolhido a partir da natureza dos dados de entrada, podendo ser desde uma Rede Convolucional (CNN), que é comumente aplicada a problemas com processamento de imagem, uma Rede Perceptron de Múltiplas

Camadas (MLP), para dados estruturados, ou uma rede recorrente como a LSTM, comumente aplicada a problemas de regressão e processamento de linguagem natural, que é o problema a ser resolvido neste estudo. No entando, a fim de gerar um comparativo entre diferentes arquiteturas base, o estudo adotou as três topologias.

O modelo LSTM base escolhido é uma arquitetura sequencial composta por uma camada de entrada contendo 28 neurônios, que é o tamanho escolhido para os termos após o pre-processamento, seguida por uma camada de *embedding*, que recebe como entrada o maior índice de vocabulário (200) e retorna uma matriz de tamanho 28 x 128, onde cada caractere dado como entrada é representado por uma lista de 128 valores. A próxima camada é a rede recorrente (LSTM), que foi configurada com 30 células de memória e funções de ativação nativas, ou seja, apenas uma função sigmóide nos filtros de esquecimento e saída, uma função sigmóide aliada a uma tangente hiperbólica no filtro de entrada e uma função tangente hiperbólica para regular o intervalo dos dados de saída. A seguinte dedica-se a evitar a ocorrência de overtraining (*overfitting*), para tanto, o modelo realiza o desligamento aleatório de 20% dos neurônios (*Dropout*). Por fim, temos uma camada de saída, que contém um número de neurônios proporcionais a quantidade de classes em análise (30 classes).

Além das três variações de arquiteturas base, foram adotadas as métricas de distância Euclidiana (p=2), Manhattan (p=1) e Minkowski (p=1,5), de forma que foram geradas nove variações de rede siamesa. As três primeiras arquiteturas (Redes 1, 2 e 3) utilizaram uma rede MLP base. Já para as Redes 3, 4, 5 o modelo base MLP foi substituído por uma rede CNN. E por fim, para as Redes 7, 8 e 9, adotou-se a arquitetura LSTM descrita anteriormente. A composição arquitetural das redes base, anteriormente descritas, foram definidas de forma empírica e podem ser observada na Tabela 1.

Tabela 1. Configurações das redes base.

Modelo Base	Hiperparâmetros
MLP	- Camadas ocultas: 3 - Neurônios por camada: 29 - Função de ativação: Tanh
CNN	- Camadas convolucionais: 2 - Quantidade de filtros: 4 e 16 - Tamanho dos filtros: 5 - Função de ativação: Tanh
LSTM	- Camadas recorrentes: 1 - Quantidade de desdobramentos: 30 - Funções de ativação: Sigmóide e Tanh

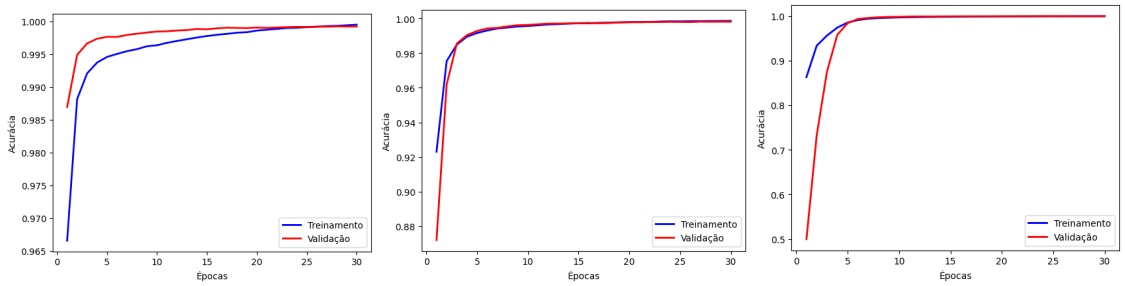
A quarta e última etapa da metodologia é a classificação do termo como genuíno ou homográfico. Nesse processo, utilizando o resultado produzido pelo modelo detector de marca-alvo, é possível inferir a autenticidade do termo que está sendo avaliado. Ao utilizar os termos genuínos como centroides para aferição de similariadade, observa-se que a distância entre este e um termo idêntico tende a zero, enquanto a distância entre o termo verdadeiro e as marcas semelhantes (*typosquatting*) varia entre valores maiores que 0,01 e menores ou iguais a 1. Portanto, podemos inferir que os termos pertencentes ao

intervalo de distância (0,01 <d <0,5), são homográficos da marca que está sendo avaliada.

4. Resultados e discussão

Esta seção apresenta e discute os resultados obtidos em experimento controlado, onde a metodologia proposta foi aplicada a um conjunto de dados composto por mais de 30.000 registros homográficos, direcionados a 30 populares marcas.

Inicialmente, foram avaliadas diversas combinações arquiteturais de redes neurais base, averiguando sua capacidade de extrair características e aproximá-las ou repeli-las a partir de uma métrica de distância. Para isso, as nove combinações de redes neurais profundas já apresentadas foram divididas em três subgrupos, permitindo combinações entre as redes neurais MLP (a), CNN (b) ou LSTM (c), e três métricas de distância. Neste cenário, a Figura 3 apresenta a curva de aprendizado para a melhor composição arquitetural de cada subgrupo. As curvas comparativas entre os conjuntos de treino e validação permitem afirmar que o ajuste dos modelos aconteceu de forma correta, demonstrando a não ocorrência de sobreajuste ou superajuste dos modelos.



(a) MLP **(b)** CNN **(c)** LSTM

Figura 3. Curva de aprendizado época-a-época.

O formato e os índices de acurácia alcançados pelas curvas de aprendizado apresentadas demonstram que as redes siamesas obtiveram sucesso na minimização da função de perca, o que indica que as redes detém a capacidade de aproximar entradas pertencentes a uma mesma marca-alvo e distanciar pares impostores. Neste sentido, a Tabela 2 apresenta a acurácia[5] média obtida por todas as combinações de redes siamesas sobre o conjunto de dados de validação. É importante ressaltar que esse conjunto foi aplicado ao ajuste do modelo, sendo assim, os resultados obtidos neste podem ser enviesados, uma vez que em um dado momento o modelo inteligente já o analisou.

Os resultados apresentados, revelam que a combinação realizada na Rede 9 apresenta a maior acurácia dentre as nove combinações, alcançando uma acurácia de 99,95%. Em contraponto, a Rede 6 apresenta o pior resultado, com uma acurácia média de 99,25%, que mesmo obtendo o menor índice, ainda é superior a 99%. Nessa observação, é notável que os resultados são proporcionais à quantidade de ciclos de treinamento, fornecendo melhores resultados para redes com mais rodadas de ajuste e piores resultados para treinamentos mais curtos.

[5]Equações para cálculo das métricas de validação: https://scikit-learn.org/stable/modules/model_evaluation.html#classification-report

Tabela 2. Acurácia percentual para os dados de teste.

	Ciclos de treinamento				
Redes	**10 épocas**	**15 épocas**	**20 épocas**	**25 épocas**	**30 épocas**
Rede 1	99,74%	99,78%	99,79%	99,77%	99,80%
Rede 2	99,84%	99,88%	99,91%	99,92%	**99,92%**
Rede 3	99,84%	99,87%	99,90%	99,91%	99,92%
Rede 4	99,62%	99,68%	99,70%	99,72%	99,73%
Rede 5	99,68%	99,74%	99,76%	99,80%	**99,82%**
Rede 6	99,59%	99,75%	99,79%	99,80%	**99,82%**
Rede 7	99,81%	99,85%	99,88%	99,88%	99,87%
Rede 8	99,86%	99,90%	99,93%	99,93%	99,92%
Rede 9	99,82%	99,91%	99,93%	99,94%	**99,95%**

Não obstante, os resultados obtidos a partir da exposição das arquiteturas a um conjunto de dados inédito (conjunto de validação), apresentados na Tabela 3, demonstram que o melhor resultado está contido no subgrupo de redes baseadas no modelo recorrente LSTM, que obteve o seu melhor índice com a Rede 9, atingiu a marca de 99,59% de taxa de acerto. Já a menor acurácia percentual manteve-se presente no subconjunto de redes baseadas na arquitetura MLP, obtendo seu pior resultado com a Rede 1, na configuração de apenas 10 rodadas de treinamento. Quando observamos apenas resultados para o volume máximo de ciclos de ajuste obtemos a configuração da Rede 4, como pior resultado, no entanto, mesmo nesse cenário, todas as configurações obtiveram mais de 90% de taxa de acerto para o nível máximo de ajuste.

Tabela 3. Acurácia percentual para os dados de validação.

	Ciclos de treinamento				
Redes	**10 épocas**	**15 épocas**	**20 épocas**	**25 épocas**	**30 épocas**
Rede 1	$91,08 \pm 2,50$	$92,76 \pm 2,76$	$91,38 \pm 3,24$	$93,02 \pm 2,84$	$93,03 \pm 2,54$
Rede 2	$98,13 \pm 1,56$	$98,34 \pm 0,93$	$98,83 \pm 0,59$	$98,84 \pm 0,65$	$99,04 \pm 0,55$
Rede 3	$98,51 \pm 1,08$	$98,64 \pm 0,76$	$98,85 \pm 0,52$	$98,85 \pm 0,81$	$98,68 \pm 0,89$
Rede 4	$78,46 \pm 2,15$	$83,88 \pm 3,98$	$86,10 \pm 3,10$	$88,71 \pm 3,13$	$90,07 \pm 2,73$
Rede 5	$92,18 \pm 2,54$	$92,98 \pm 2,80$	$94,32 \pm 2,03$	$95,82 \pm 1,71$	$96,06 \pm 1,82$
Rede 6	$95,00 \pm 1,98$	$95,71 \pm 1,93$	$96,90 \pm 1,43$	$97,56 \pm 1,02$	$97,77 \pm 1,01$
Rede 7	$96,26 \pm 2,08$	$98,40 \pm 0,26$	$98,29 \pm 0,52$	$98,09 \pm 0,85$	$98,39 \pm 0,45$
Rede 8	$98,77 \pm 0,48$	$99,32 \pm 0,14$	$99,38 \pm 0,09$	$99,41 \pm 0,13$	$99,43 \pm 0,13$
Rede 9	$99,41 \pm 0,14$	$99,57 \pm 0,11$	$99,59 \pm 0,08$	$99,58 \pm 0,11$	$99,55 \pm 0,10$

Outro ponto é que, diferentemente do conjunto de validação, para os registros dos dados de teste, a Rede 9 alcançou seu melhor resultado na época 20, demonstrando que essa configuração pode ser eficiente com um menor volume de ajuste, o que, consequentemente, permite um treinamento mais eficiente e rápido. Mais um ponto a observar, é a dificuldade dos modelos baseados em redes MLP em realizarem a extração de características capazes de generalizar as entradas, o que é evidenciado com a presença, tanto para o conjunto de teste quanto para o de validação, do pior resultado em suas configurações.

Ainda sobre a composição das redes, também fica evidente a desvantagem na aplicação da Distância Euclidiana como métrica de similaridades. Observando os resul-

tados para o conjunto de teste, nota-se que, para todos os cenários, as combinações que utilizam essa métrica apresentam os piores resultados internos ao subgrupo, o que não é diferente quando observamos os resultados de forma geral, já que as três configurações com essa característica apresentam os piores resultados.

Outro paradigma a analisar, acerca das acurácias do conjunto de testes, é o desvio padrão. Dentre as vinte execuções dos modelos, os resultados mostram que a Rede 9, com 20 ciclos de ajuste apresenta o menor desvio interno ao seu subgrupo, com a marca de 0,08%, demonstrando a constância da arquitetura, que mesmo sendo treinada com diferentes conjuntos de dados, mantém um baixo distanciamento da média do resultado.

Para a validação das redes, também foi escolhido como métrica o F1-score. Esse mecanismo permite a análise conjunta das medidas de precisão e revocação através de uma média harmônica entre os resultados. Os resultados desta métrica denotam a capacidade global dos modelos em identificar um par de entradas tal como requisitado. Como resultados para essa métrica, obtivemos os valores presentes na Tabela 4, que reforçam a já observada vantagem entre dos modelos baseados em redes LSTM agregadas a métrica de similaridade de Mikowski.

Tabela 4. F1-*score* percentual para os dados do conjunto de validação.

	Ciclos de treinamento				
Redes	**10 épocas**	**15 épocas**	**20 épocas**	**25 épocas**	**30 épocas**
Rede 1	89,43%	90,79%	89,65%	91,31%	91,03%
Rede 2	95,40%	95,48%	95,94%	95,94%	**96,08%**
Rede 3	95,47%	95,68%	95,91%	95,85%	95,77%
Rede 4	77,92%	82,94%	84,78%	87,32%	88,66%
Rede 5	90,40%	91,03%	92,35%	93,63%	93,83%
Rede 6	92,34%	93,23%	94,32%	94,84%	**95,06%**
Rede 7	94,24%	95,87%	95,79%	95,62%	95,88%
Rede 8	95,85%	96,27%	96,33%	96,36%	96,39%
Rede 9	96,23%	96,39%	96,43%	**96,44%**	96,43%

Observada a eficiência da rede siamesa, iniciamos a avaliação da capacidade de identificação da marca-alvo. Nesse processo, utilizamos a capacidade da rede neural treinada para identificar a marca para qual o termo homográfico faz referência. Sendo assim, adotamos o rótulo genuíno das marcas-alvo como centróides, para aferição de distância para os termos homográficos desta forma, o termo que possuir características com menor distância para um centróide, é classificado como pertencente a essa marca.

A métrica selecionada para composição dos resultados da identificação de marca foi a precisão. Através dela, é possível observar a eficiência percentual das diversas composições de rede neural. Iniciando pela precisão, as Tabelas 5 e 6, apresentam a precisão percentual das arquiteturas para a identificação das marcas-alvo ”Bank of America”e ”TSB Bank”, que obtiveram os melhores e piores resultados percentuais.

Com os resultados, é possível notar que a performance dos modelos, em especial as composições baseadas em redes MLP e CNN, variam com maior amplitude para marcas que possuem menores quantidades de caracteres em seus rótulos, o que reforça as observações anteriores quanto a dificuldade de tais métricas em extrair características

Tabela 5. Precisão percentual para os registros da marca #12 (Bank of America).

	Ciclos de treinamento				
Redes	**10 épocas**	**15 épocas**	**20 épocas**	**25 épocas**	**30 épocas**
Rede 1	$99,16 \pm 0,88$	$99,56 \pm 0,73$	$99,73 \pm 0,40$	$99,73 \pm 0,38$	$99,79 \pm 0,29$
Rede 2	$99,64 \pm 0,33$	$99,73 \pm 0,40$	$99,75 \pm 0,40$	$99,75 \pm 0,35$	$99,75 \pm 0,40$
Rede 3	$99,82 \pm 0,30$	$99,77 \pm 0,31$	$99,71 \pm 0,65$	$99,62 \pm 0,74$	$99,66 \pm 0,34$
Rede 4	$98,29 \pm 0,82$	$99,04 \pm 0,82$	$99,09 \pm 0,75$	$99,25 \pm 0,68$	$98,75 \pm 1,54$
Rede 5	$99,38 \pm 0,34$	$99,50 \pm 0,45$	$99,54 \pm 0,45$	$99,38 \pm 0,55$	$99,27 \pm 0,70$
Rede 6	$99,45 \pm 0,62$	$99,39 \pm 0,59$	$99,46 \pm 0,60$	$99,32 \pm 0,62$	$99,41 \pm 0,47$
Rede 7	$98,88 \pm 0,72$	$99,13 \pm 0,56$	$99,22 \pm 0,81$	$99,22 \pm 0,72$	$99,25 \pm 0,78$
Rede 8	$99,54 \pm 0,55$	$99,72 \pm 0,39$	$99,73 \pm 0,38$	$99,75 \pm 0,42$	$99,86 \pm 0,24$
Rede 9	$99,89 \pm 0,20$	$99,87 \pm 0,18$	$99,68 \pm 0,56$	$99,95 \pm 0,13$	$99,73 \pm 0,42$

Tabela 6. Precisão percentual para os registros da marca #16 (TSB Bank).

	Ciclos de treinamento				
Redes	**10 épocas**	**15 épocas**	**20 épocas**	**25 épocas**	**30 épocas**
Rede 1	$72,84 \pm 34,44$	$60,05 \pm 43,22$	$56,49 \pm 39,76$	$66,21 \pm 36,99$	$66,96 \pm 37,69$
Rede 2	$96,19 \pm 5,10$	$96,98 \pm 4,65$	$92,57 \pm 18,27$	$96,39 \pm 10,73$	$95,62 \pm 15,88$
Rede 3	$95,04 \pm 10,97$	$97,94 \pm 1,52$	$97,17 \pm 2,48$	$98,76 \pm 0,90$	$95,32 \pm 14,77$
Rede 4	$13,41 \pm 15,93$	$16,43 \pm 20,31$	$26,07 \pm 28,82$	$50,08 \pm 40,83$	$27,32 \pm 33,73$
Rede 5	$59,43 \pm 28,89$	$58,61 \pm 32,76$	$73,14 \pm 26,15$	$76,44 \pm 29,16$	$72,85 \pm 28,29$
Rede 6	$70,90 \pm 28,34$	$78,63 \pm 26,22$	$87,87 \pm 19,08$	$95,23 \pm 2,83$	$88,64 \pm 22,32$
Rede 7	$85,08 \pm 25,30$	$97,52 \pm 1,13$	$97,45 \pm 1,15$	$97,50 \pm 1,63$	$97,35 \pm 1,43$
Rede 8	$93,34 \pm 13,33$	$98,59 \pm 1,19$	$98,81 \pm 0,87$	$98,71 \pm 0,96$	$98,79 \pm 0,73$
Rede 9	$97,40 \pm 1,80$	$98,74 \pm 0,84$	$98,98 \pm 1,05$	$99,13 \pm 0,58$	$99,13 \pm 0,58$

capazes de distinguir entre as marcas participantes. Em sentido oposto, as arquiteturas compostas por redes recorrentes, apresentaram pequenas reduções de precisão quando apresentadas a entradas com menos caracteres, além de demonstrarem um aumento em relação à acurácia total dos modelos.

Outro ponto a observar, é o desvio percentual dos modelos após as rodadas de teste. Os resultados demonstram a fragilidade dos modelos dos subgrupos 1 e 2, que obtiveram desvios com máximos superiores a 30%, evidenciando a sensibilidade do modelo ao conjunto de dados de treino. Dentre as Redes compostas pela arquitetura recorrente LSTM, é notável que a redução do desvio está relacionada ao aumento no volume de ciclos de treinamento, tendo alcançado o menor percentual na composição entre a arquitetura LSTM e a Distância de Minkowski.

5. Ameaças e limitações

Esta subseção apresentará algumas ameaças e limitações, tanto do motor para elaboração dos termos homográficos, quanto do modelo para detecção dos ataques.

5.1. Ameaças e limitações dos modelos

Iniciando pelas limitações empregadas na construção da base de dados: as características aplicadas à deturpação dos termos genuínos são populares na concepção de *typosquatting*.

No entanto, com o avançar do tempo, novos padrões de violação podem surgir, evidenciando a necessidade do acompanhamento contínuo destas, para que, quando evidenciadas, possam ser aplicadas ao motor. Quanto às características já incluídas no modelo, algumas detém peculiaridades que podem causar limitações, como por exemplo a característica #02, que aplica o plural ao termo genuíno. As normas para conversão de um termo em seu plural são pautadas em regras gramaticais oriundas de sua língua, sendo assim, como não fez parte do escopo desse projeto a identificação do idioma a qual o termo pertence, existe a significativa possibilidade de que a manipulação acabe por provocar uma deturpação que não corresponde a característica solicitada.

Ainda sobre a base de dados, o motor só detém a capacidade de construção de violações do tipo *typosquatting* e não de termos *cybersquatting*. A modalidade caracterizada pela aplicação de termos precedidos de grafia correta, não permite a elaboração de homográficos artificiais. Apesar de ter contado com o rótulo genuíno para representação de tal modalidade de ataque, o que é observado em cerca de 70% das ocorrências com essa topologia homográfica, será necessária a construção manual de uma base de dados com essa característica.

Sobre a adaptação parcial do modelo a algumas topologias homográficas, a análise de termos em *Unicodes* de outros idiomas, por exemplo o cirílico, também não fez parte do escopo do trabalho. Dessa forma, quando aplicado a um cenário real, o modelo não irá identificar referências a uma marca, quando esta apresentar-se em tais *Unicodes*.

5.2. Limitações do experimento

O primeiro ponto a observar é o alto desequilíbrio entre a quantidade de termos homográficos por característica. As regras adotadas para construção da base de dados aplicada aos experimentos acabaram por gerar uma grande variação entre a quantidade de termos gerados a partir das características. Quando comparamos os resultados para o comportamento #02 e #10, menor e maior quantidade de termos deturpados artificiais, podemos observar o tamanho da discrepância entre os resultados, que foram 1 para o primeiro comportamento e 892 para o segundo, o que pode causar um enviesamento do modelo para identificação da característica com maior volume de dados.

Outra limitação pode ser observada na construção dos termos pertencentes às características #01 e #06. Para execução desse experimento, adotou-se os padrões de teclado ”QWERTY”, ”AZERTY”, ”DVORAK”e ”COLEMAK”, que figuram como os mais populares do mundo, no entanto, existe uma variedade de modelos alternativos com aplicação em menor escala, a exemplo de ”MALTRON”, ”JCUKEN”e ”NEO”. A escolha de alguns padrões de teclado em detrimento de outros expõe uma vulnerabilidade do experimento, uma vez que, em ambiente de produção, pode haver termos deturpados a partir de padrões de teclado inéditos para o modelo de detecção.

Quanto ao mecanismo adotado para classificação dos termos, para os experimentos presentes neste estudo, adotou-se a distância entre o termo em análise e centróides, que foram definidos como o rótulo genuíno. No entanto, quando aplicado ao ambiente real e atuação *phishing*, tal mecanismo será ineficaz, visto que o rótulo da marca, precedido de grafia correta, pode sugerir que a página é genuína. Uma vez que este poderá se apresentar em diversos pontos da URL e conteúdos da página em análise, assim como acompanhado de inúmeros termos genéricos.

6. Conclusão

O trabalho apresentou o desenvolvimento de um sistema inteligente, capaz de identificar ataques homográficos que replicam comportamentos observado em páginas *phishing*, que exploram critérios de fidedignidade para conferir autenticidade ao ataque. Os resultados apresentados indicam que todas as arquiteturas testadas atingem um bom índice de assertividade, quando ajustadas por um período mínimo de 15 ciclos de treinamento. Dentre as combinações testadas, destacam-se as arquiteturas compostas por redes neurais LSTM, que obtiveram acurácia superior a 90% em todas as combinações aplicadas.

Quanto às métricas de distância e similaridade, a medida de Minkowski alcançou os melhores resultados, em especial quando combinada com a arquitetura LSTM. Essa composição obteve uma acurácia média de 99,59% com um desvio padrão de 0,08%, quando ajustada por 20 ciclos de treinamento. No entanto, faz-se necessária a aplicação das arquiteturas a testes mais complexos, como por exemplo a aplicação dos modelos a termos homográficos de classes inéditas.

Observando os resultados na perspectiva da identificação de marca, a composição formada entre a rede base LSTM e distância de Minkowski, demonstrou uma melhor capacidade de extração de características para entradas com menos caracteres em sua composição, fortalecendo seu favoritismo dentre as demais composições. Quando comparada a outras soluções presentes na literatura, os resultados demonstraram que a composição de Rede 9, supera diversas ferramentas, a exemplo da *TypoPegging*, que é baseada em distância de edição e atingiu 93,43% de acurácia, e outras arquiteturas inteligentes, como *TypoEval*, que obteve 97,19% de acurácia.

Diante da observada expertise da arquitetura na identificação de termos homográficos em páginas *phishing*, e o alto impacto social oriundo das atuações bem sucedidas dos ataques, relatamos aqui algumas sugestões para aprimoramento da ferramenta. A primeira sugestão envolve a validação do estudo em ambiente real, ou seja, expor as arquiteturas a entradas oriundas de páginas *phishing* reais, que podem apresentar peculiaridades não observadas em ambiente simulado. Ainda nessa temática, sugerimos a composição de uma base de dados complementar, povoada por termos diferentes dos rótulos das marcas-alvo, mas que fazem referência a esta, como por exemplo sinônimos, abreviaturas, palavras reservadas, entre outros.

Por fim, sugerimos a utilização de outras variações de rede neural recorrente, a exemplo da rede *Gated Recurrent Unit* (GRU), que pode demonstrar a mesma eficiência das apresentadas, mas com menor custo.

Referências

Ahmad, I., Parvez, M. A., and Iqbal, A. (2019). Typowriter: A tool to prevent typosquatting. In *2019 IEEE 43rd COMPSAC*, volume 1, pages 423–432.

Buber, E., Demir, O., and Sahingoz, O. K. (2017). Feature selections for the machine learning based detection of phishing websites. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–5.

Chiba, D., Akiyama, M., Yagi, T., Hato, K., Mori, T., and Goto, S. (2018). Domainchroma: Building actionable threat intelligence from malicious domain names. *Computers & Security*, 77:138–161.

Dam, T., Klausner, L. D., Buhov, D., and Schrittwieser, S. (2019). Large-scale analysis of pop-up scam on typosquatting urls. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19. ACM.

Gilwit, D. (2003). The latest cybersquatting trend: Typosquatters, their changing tactics, and how to prevent public deception and trademark infringement. *The Journal of Law and Policy*.

Le Pochat, V., Van Goethem, T., and Joosen, W. (2019). A smörgåsbord of typos: Exploring international keyboard layout typosquatting. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 187–192.

Liu, T., Zhang, Y., Shi, J., Jing, Y., Li, Q., and Guo, L. (2016). Towards quantifying visual similarity of domain names for combating typosquatting abuse. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 770–775.

Moubayed, A., Injadat, M., Shami, A., and Lutfiyya, H. (2018). Dns typo-squatting domain detection: A data analytics & machine learning based approach. In *2018 IEEE Global Communications Conference (GLOBECOM)*, page 1–7. IEEE.

Piredda, P., Ariu, D., Biggio, B., Corona, I., Piras, L., Giacinto, G., and Roli, F. (2017). Deepsquatting: Learning-based typosquatting detection at deeper domain levels. In *Conference of the Italian Association for Artificial Intelligence*, pages 347–358.

Quinkert, F., Lauinger, T., Robertson, W., Kirda, E., and Holz, T. (2019). It's not what it looks like: Measuring attacks and defensive registrations of homograph domains. In *2019 IEEE Conference on Communications and Network Security (CNS)*.

Spaulding, J., Nyang, D., and Mohaisen, A. (2017). Understanding the effectiveness of typosquatting techniques. In *Proceedings of the Fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies*, HotWeb '17. ACM.

Spaulding, J., Upadhyaya, S., and Mohaisen, A. (2016). The landscape of domain name typosquatting: Techniques and countermeasures. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pages 284–289.

Tahir, R., Raza, A., Ahmad, F., Kazi, J., Zaffar, F., Kanich, C., and Caesar, M. (2018). It's all in the name: Why some urls are more vulnerable to typosquatting. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 2618–2626.

Teixeira, L., Silva, C., Fernandes, B., Oliveira, J., Feitosa, E., Filho, G. C., Arcoverde, H., and Garcia, V. (2021). Uma avaliação de comportamentos homográficos em ataques de phishing direcionados que exploram a suscetibilidade pela fidedignidade e sazonalidade. In *Anais do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 253–266, Porto Alegre, RS, Brasil. SBC.

Ya, J., Liu, T., Li, Q., Lv, P., Shi, J., and Guo, L. (2018). Fast and accurate typosquatting domains evaluation with siamese networks. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 58–63.

Zhu, W., Yao, T., Ni, J., Wei, B., and Lu, Z. (2018). Dependency-based siamese long short-term memory network for learning sentence representations. *PLOS ONE*, 13(3):1–14.