Improving the Adversarial Robustness of Machine Learning-based Phishing Website Detectors: An Autoencoder-based Auxiliary Approach

Yang Gao
Indiana University
gaoyang@iu.edu

Sagar Samtani
Indiana University
ssamtani@iu.edu

Ankit Shah
Indiana University
ankit@iu.edu

Abstract

Anti-phishing research relies on collaboration between defensive and offensive efforts. The defensive side develops machine learning-based phishing website detectors to protect users from phishing attacks. However, adversaries can manipulate detectable phishing websites into evasive ones as adversarial examples, misleading detectors into classifying them as legitimate. Therefore, offensive efforts are vital to examine the threats posed by adversaries and inform the defensive side to improve the adversarial robustness of detectors. Prevailing approaches to improve adversarial robustness may compromise a detector's original high performance on clean data (non-adversarial websites) as it becomes more accurate at detecting adversarial examples. To address this, we propose a novel approach using a Graph Convolutional Autoencoder as an auxiliary model to make collaborative decisions with the original detector in distinguishing evasive phishing websites from legitimate ones. We evaluate our approach by enhancing a CNN-based detector against adversarial attacks. Our approach achieves high adversarial robustness while maintaining high performance on clean data compared to retraining and fine-tuning benchmarks.

Keywords: Cybersecurity, Machine learning, Adversarial robustness, Phishing website detection, Graph convolutional autoencoder

1. Introduction

Phishing attacks are a prevalent cybercrime, with phishing websites being a particularly grave vector. Phishing websites are legitimate-looking web pages to trick unsuspecting users into divulging vital information, such as usernames, passwords, credit card numbers, and other personal details (Tian et al., 2018). In recent years, many anti-phishers have turned to machine learning to build phishing website detectors. Prior work has demonstrated the capabilities of machine learning (ML) models to detect phishing websites automatically (Abbasi et al, 2015; Ma et al., 2009; Opara et al., 2020; Ouyang and Zhang, 2021; Smadi et al., 2018). However, ML-based detectors can be evaded by adversarial examples generated by adversarial attacks. Attackers can carefully manipulate phishing websites using evasion techniques to mislead less robust detectors into classifying phishing as benign (Apruzzese et al., 2022; Montaruli et al., 2023). The deployment of ML-based detectors lacking sufficient robustness against such attacks could result in many phishing websites bypassing defense mechanisms, ultimately reaching web users and causing substantial losses. Therefore, it is crucial to make ML-based phishing website detectors more robust to adversarial attacks.

To improve adversarial robustness, it is vital for offensive side researchers to study adversarial threats and inform the defensive side to effectively incorporate the knowledge of adversarial examples into their defense models. Existing studies on adversarial robustness in ML-based phishing website detectors have several shortcomings. First, the prevailing approach to enhance adversarial robustness in phishing contexts is to retrain or fine-tune the detector using adversarial examples generated by adversarial attacks. However, there exists a trade-off between clean data (non-adversarial) and adversarial data for a single ML model (Wang et al., 2020). Shifting the weights of the detector to enhance its ability to detect adversarial examples can reduce its accuracy on clean data. Conversely, prioritizing performance on clean data may limit improvements against adversarial attacks. Second, those studies aiming to emulate the most realistic adversarial attacks solely focus on the offensive perspective to demonstrate that ML-based phishing website detectors are vulnerable. However, they fall short in proposing mitigation strategies to enhance the detectors against adversarial attacks. Studies focusing on the defensive side often assume theoretical adversarial attacks as threat models, increasing the detector's robustness. Such approaches adopt gradient-based threat models from the

URI: https://hdl.handle.net/10125/108886
978-0-9981331-8-8

image classification context, which fails to represent the realistic threat posed by adversarial phishing websites.

In our study, we design an adversarial robustness approach that collaboratively considers the defensive and offensive perspectives of anti-phishing research. This approach enhances the robustness of ML-based phishing website detectors against realistic adversarial attack scenarios and addresses the performance trade-off between clean data and adversarial examples. The proposed approach has two major components:

- First, our offensive-side work deploys a threat model that emulates adversaries' evasion techniques to generate well-crafted phishing websites as adversarial examples, which can assess the robustness of detectors.
- Second, informed by the insights gained from offensive-side work, we design a Graph Convolutional Autoencoder (GCAE)-based auxiliary model that can be added to the original detector to filter out adversarial examples. It works as a second gate of detection without altering anything within the well-trained original detector.

The collaborative decision-making between the original detector and the auxiliary model mitigates the original detector's vulnerability to adversarial examples without compromising its ability to detect non-adversarial phishing and legitimate websites.

The remainder of this paper is organized as follows. First, we review literature related to adversarial robustness and graph convolutional autoencoders. Second, we identify research gaps and pose research questions for study. Third, we present our proposed framework and its constituent components. Fourth, we present our evaluation and, subsequently, our results. Finally, we summarize our contributions and conclude this research.

2. Literature Review

We review two areas of literature to set the foundation of our research. First, we examine prior studies on adversarial robustness in phishing website detection, considering both the offensive and defensive perspectives. Second, we review studies on autoencoders that use reconstruction error for anomaly detection and explore how this principle is applied in detecting adversarial examples.

2.1. Adversarial Robustness

Achieving adversarial robustness requires understanding both the offensive and defensive perspectives. Work concentrating on the offensive side seeks to develop threat models that emulate the risks posed by adversaries to attack the defensive mechanisms. Defensive side studies focus on developing robustness approaches that improve detection accuracy on adversarial examples and raise the attack cost based on understanding the offensive side.

2.1.1. Offensive-side Studies focus on evading ML-based phishing website detectors through adversarial attacks, in which adversarial examples that can bypass detection are generated. The foundational research on adversarial attacks and robustness originates from image tasks (Bai et al., 2021; Goodfellow et al., 2014). Thus, many adversarial attacks in the phishing context apply the same theoretical attacks used in image classification tasks. These threat models introduce noise to the feature vectors extracted by detectors to mislead detection, such as using Generative Adversarial Networks (GAN) (O'Mara et al., 2021; Shirazi et al., 2021). However, such threat models are poorly suited for the context of phishing. Adding noise to the feature vectors of images, which are representations of pixels, can be directly reflected in the images, resulting in new images as adversarial examples. In contrast, adding noise to the feature vectors of phishing websites cannot be mapped back to real evasion techniques and cannot be reversed back to well-rendered phishing websites (Montaruli et al., 2023). This indicates that such theoretical adversarial attacks are far from realistic in the context of phishing, leading to mitigations based on unrealistic threat assumptions. For example, Shirazi et al. (2021) enhanced ML-based detectors that extract features from HTML source code by using feature vectors generated from unrealistic threat models.

To address this issue, some studies have focused on emulating phishing evasion attacks as realistically as possible. These studies developed threat models that use evasion techniques to manipulate the source code of phishing websites, generating well-rendered evasive phishing websites as adversarial examples (Apruzzese et al., 2022; Montaruli et al., 2023; Song et al., 2021). Common evasion techniques include injection and obfuscation, which can inject invisible benign content or hide phishing content within the source code to mislead detectors. These studies significantly contribute to evaluating the vulnerabilities of ML-based phishing website detectors by posing realistic threats in the phishing context. However, they only demonstrate the threats posed by their attacks without proposing any mitigation methods, raising significant concerns about detecting such real adversarial attacks.

2.1.2. Defensive-side Studies often seek to develop ML-based phishing website detectors that heavily rely on URL, HTML, and DOM (created by parsing HTML) to extract domain knowledge-based features or embeddings. Both traditional machine learning classifiers such as Logistic Regression, Bayesian Network, J48 Decision Tree, Support Vector Machine, Random Forest, Adaboost and deep learning classifiers such as Convolutional Neural Network (CNN) and GCN (Graph Convolutional Network) are commonly adopted as classifiers (Abbasi et al, 2015; Opara et al., 2020; Ouyang and Zhang, 2021; Xiang et al., 2011).

When an ML-based phishing website detector is found to lack robustness against adversarial attacks, defensive research focuses on enhancing the adversarial robustness of the detector based on the understanding of the threat models. In many defensive studies, threat models generate adversarial examples from the phishing samples in the detector's training set, providing new training samples for adversarial robustness. These adversarial examples are then used to retrain or fine-tune the detector to recognize how adversarial examples can deviate from the detectable phishing samples (Mehdi Gholampour and Verma, 2023; Sabir et al., 2022; Shirazi et al., 2021).

Retraining and fine-tuning have significant limitations in that they shift the weights of the original detector. In retraining, the training set of the original detector is augmented with the generated adversarial examples, and the detector is retrained from scratch on the augmented training set. In fine-tuning, the generated adversarial examples are used to slightly adjust the weights of the pre-trained detector. However, it has been demonstrated that there is a trade-off between clean data (non-adversarial) and adversarial examples for a single machine-learning model (Wang et al., 2020). Improving adversarial robustness might hurt its detection accuracy on clean data while maintaining detection accuracy on clean data might limit the improvement of adversarial robustness. No matter how a single detector tries to balance these two aspects, relying solely on one detector makes achieving optimal performance in both aspects difficult. Therefore, we need a novel adversarial robustness method that can maintain the high accuracy of the original detector on clean data while mitigating its vulnerability to adversarial examples. Instead of adjusting a single detector, one potential mechanism is introducing an auxiliary model to filter out adversarial examples once they bypass the original detector.

2.2. Graph Convolutional Autoencoder

Autoencoders are particularly effective due to their ability to learn data representations and reconstruct inputs, making them suitable for distinguishing a specific class from a mixture of multiple types of samples in datasets. When trained on a specific class of samples, autoencoders can learn to reconstruct these samples well, thereby filtering out samples that do not belong to that class (Ma et al., 2021). This makes an autoencoder trained on adversarial examples an appropriate auxiliary model for distinguishing adversarial examples from legitimate websites when they are mixed together. In the context of phishing website detection, realistic adversarial examples are well-rendered, evasive phishing websites with carefully manipulated HTML source code (Montaruli et al., 2023). The auxiliary model needs to learn the unique HTML patterns of these examples.

HTML source code is organized by elements such as <html>, <body>, <div>, <p>, etc. The elements are linked together, creating parent-child and sibling relationships. Therefore, Graph Convolutional Autoencoder (GCAE) is particularly effective for processing HTML source code because it has been shown to perform well in learning source code representations in a syntax tree structure (Ding et al., 2023). GCAEs combine the strengths of autoencoders and graph convolutional networks to process data structured as graphs. The input to a GCAE is a graph $G = (V, E)$ where V is the set of nodes, and E is the set of edges. Each node v_i is associated with a feature vector x_i. The encoder of the GCAE consists of multiple graph convolutional layers that progressively reduce the dimensionality of the input features. After the encoding layer, the input graph is represented in a lower-dimensional latent space z that captures the most important features of the input graph. The decoder of the GCAE attempts to reconstruct the original input graph from the latent representation. In the context of HTML source code, elements are represented as nodes, and parent-child and sibling relationships between elements are represented as edges.

3. Research Questions

We identified several limitations from the prior adversarial robustness studies on phishing website detection. First, there is a lack of work on enhancing the adversarial robustness of ML-based phishing website detectors against realistic adversarial attacks, where well-rendered phishing websites are generated as adversarial examples to evade detection. Second, commonly used adversarial robustness methods, such as retraining or fine-tuning ML-based phishing website detectors, face a trade-off between detection accuracy on clean data and adversarial examples. Based on these limitations, we pose the following research questions for the study:

- How can we enhance the adversarial robustness of the detector against threat models that generate well-rendered phishing websites as adversarial examples?

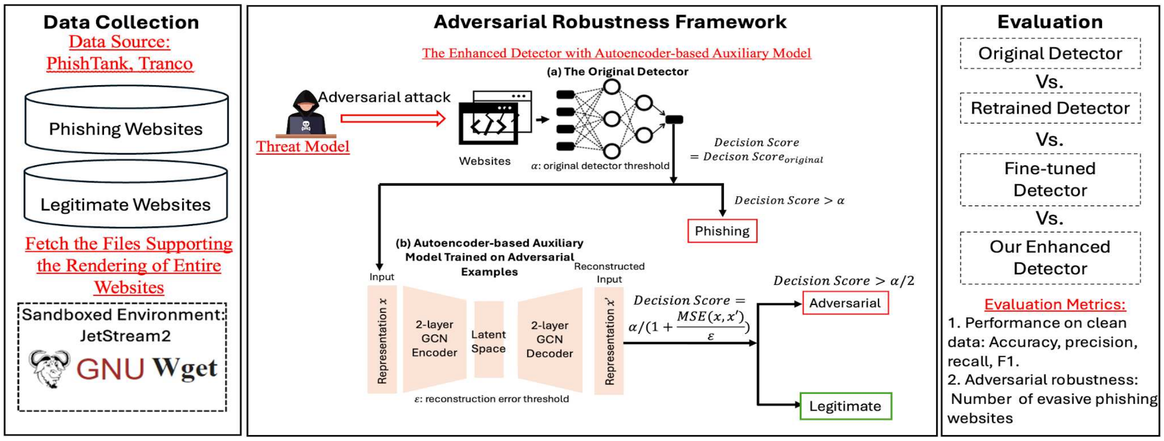
Data Collection
Data Source:
PhishTank, Tranco
Phishing Websites
Legitimate Websites
Fetch the Files Supporting the Rendering of Entire Websites
Sandboxed Environment:
JetStream2
GNU Wget
Adversarial Robustness Framework
The Enhanced Detector with Autoencoder-based Auxiliary Model
(a) The Original Detector
Adversarial attack
Threat Model
Websites
α: original detector threshold
Decision Score = Decison Score_original
Decision Score > α
Phishing
(b) Autoencoder-based Auxiliary Model Trained on Adversarial Examples
Input
Representation x
2-layer GCN Encoder
Latent Space
2-layer GCN Decoder
Reconstructed Input
Representation x'
ε: reconstruction error threshold
Decision Score = α/(1 + MSE(x,x')/ε)
Decision Score > α/2
Adversarial
Legitimate
Evaluation
Original Detector
Vs.
Retrained Detector
Vs.
Fine-tuned Detector
Vs.
Our Enhanced Detector
Evaluation Metrics:
1. Performance on clean data: Accuracy, precision, recall, F1.
2. Adversarial robustness: Number of evasive phishing websites

Figure 1. Adversarial Robustness Research Design.

- How can we adapt a GCAE as an auxiliary model to the original detector, improving the detection of adversarial examples without compromising the detector's accuracy on clean data?

4. Proposed Research Design

To address our research questions, we propose a novel adversarial robustness with three components (Figure 1): (1) Data collection; (2) An adversarial robustness framework with a threat model providing realistic emulation of adversarial attacks and a GCAE as an auxiliary model to enhance the original detector; (3) Evaluation, where we compare the detection accuracy on clean data and the robustness to adversarial attacks for the original detector, retrained detector, fine-tuned detector, and the proposed approach.

4.1. Data Collection

We developed a web crawler to harvest phishing and legitimate websites to train detectors for our experiment. The crawler utilized GNU Wget (Dobolyi and Abbasi, 2016; Purwanto et al., 2023) to visit URLs and retrieve and download all files necessary to reproduce the rendering of websites offline. The collection was conducted in a sandboxed environment on Jetstream2 (Hancock et al., 2021). We ran the crawler on May 06, 2024, to collect verified phishing websites from PhishTank, an online phishing verification platform, and obtained 10,000 unique phishing websites, each capable of rendering a unique appearance offline. Similarly, we fetched 10,000 legitimate websites from Tranco (Pochat et al., 2019), a research-oriented top sites ranking list that provides a list of legitimate websites with malicious sites removed. In total, we obtained 20,000 clean (non-adversarial) data points for the training and test set of detectors.

4.2. Adversarial Robustness Framework

Our adversarial robustness framework has two components. The offensive side is a threat model that manipulates the HTML source code of detectable phishing websites, transforming them into evasive phishing websites as adversarial examples without breaking the rendering. The defensive side is an enhanced version of the detector, where we design an autoencoder-based auxiliary model to filter out adversarial examples missed by the original detector.

4.2.1. Threat Model. Given the high realism of adversarial attacks proposed by Montaruli et al. (2023), we leverage their algorithms as threat models, attacking our detectors to generate well-rendered phishing websites as adversarial examples. Montaruli et al. (2023) designed a query-efficient black-box optimization algorithm to optimally employ 16 evasion techniques (Table 1) to manipulate the HTML code of detectable phishing websites, minimizing the decision score at which the target detector can classify phishing websites. Each manipulation on a phishing website produces a new version of the HTML source code without changing its rendering. All evasion techniques selected are frequently used by adversaries in the wild, such as injecting invisible internal and external links, adding fake copyright, obfuscating JavaScript, and altering CSS styles of hidden elements to be less detectable. The adversarial attacks can be deployed in black-box scenarios, requiring no information about the detector's training data, features, or architectures. The threat model only requires the classification result – the decision score from the target detector - following each manipulation as the feedback to judge the effectiveness of its manipulations.

Table 1. Evasion Techniques of Threat Model.

Category	Evasion Technique	Manipulation method
Injection	InjectIntElem	Injecting 10 internal links by <a> elements in body/head.
	InjectIntElemFoot	Injecting 10 internal links by <a> elements in the footer.
	InjectIntLinkElem	Injecting 10 internal links by <link> elements.
	InjectExtElem	Injecting 10 external links by <link> elements in body/head.
	InjectExtEleFoot	Injecting 10 external links by <link> elements in the footer.
	InjectFakeFavicon	Injecting a fake favicon by <link> element.
	InjectFakeCopyright	Injecting a fake copyright by <p> element.
Obfuscation	ObfuscateExtLinks	Replacing external links in the body's elements with invalid internal links and putting external links back by JavaScript at execution time.
	ObfuscateJS	Encoding JavaScript function within a <script> element.
Updating Sensitive Information	UpdateForm	Replacing action's attribute value of forms.
	UpdateIntAnchors	Replacing href's attribute value of anchors.
	UpdateHiddenDivs	Replacing style's attribute value of <div> elements.
	UpdateHiddenButtons	Removing the "disabled" attribute of <button> elements and putting the "disabled" attribute back by JavaScript at execution time.
	UpdateHiddenInputs	Replacing type attribute value of <input> elements or removing the "disabled" attribute of <input> elements and putting the "disabled" attribute back by JavaScript at execution time.
	UpdateIframe	Replacing style's attribute value of <iframe> elements.
	UpdateTitle	Replacing the original title with domain name and putting the original title back by JavaScript at execution time.

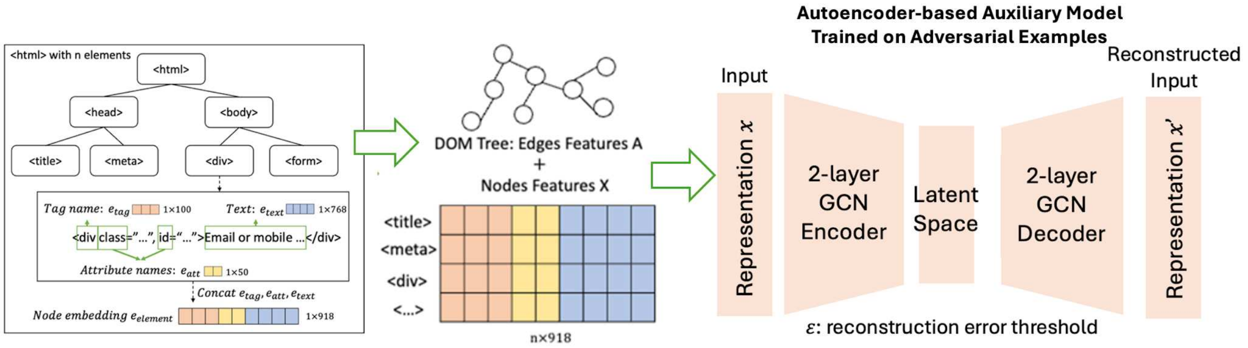
<html> with n elements
<html>
<head>
<body>
<title>
<meta>
<div>
<form>
Tag name: e_{tag} 1×100
Text: e_{text} 1×768
<div class="…", id="…">Email or mobile …</div>
Attribute names: e_{att} 1×50
Concat $e_{tag}, e_{att}, e_{text}$
Node embedding $e_{element}$ 1×918
DOM Tree: Edges Features A
+
Nodes Features X
<title>
<meta>
<div>
<…>
n×918
Autoencoder-based Auxiliary Model
Trained on Adversarial Examples
Input
Representation x
2-layer GCN Encoder
Latent Space
2-layer GCN Decoder
Reconstructed Input
Representation x'
ε: reconstruction error threshold

Figure 2. Autoencoder-based Auxiliary Model.

4.2.2. Enhanced Detector with Autoencoder-based Auxiliary Model. The enhanced detector consists of two components: (a) an original detector well-trained on clean data without considering adversarial attacks and (b) an autoencoder-based auxiliary model that filters out adversarial examples misclassified as legitimate by the original detector. Our threat model and the autoencoder-based auxiliary model are independent of any specific design of the original detector. Therefore, our adversarial robustness framework can enhance any detector vulnerable to adversarial attacks that manipulate HTML source code as long as the detector extracts part of its features from HTML.

Typically, an ML-based phishing website detector produces prediction probability as the decision score and has a threshold based on the decision score to determine the predicted class label. In the original detector, we use $DS_{original}$ to denote its decision score and α to denote the threshold. The $DS_{original}$ ranges from 0 to 1. The original detector classifies a website as phishing if $DS_{original} > \alpha$ and as legitimate if $DS_{original} \leq \alpha$. In adversarial attacks, adversaries manipulate phishing websites to cause the original detector to misclassify them. This means that the $DS_{original}$ for generated adversarial examples will be intentionally reduced to within the range $(0, \alpha)$. After an attack, relying solely on $DS_{original} \leq \alpha$ to classify legitimate websites cannot distinguish adversarial examples from true legitimate websites. Therefore, to address this issue, we design an autoencoder-based auxiliary model to differentiate adversarial examples from true legitimate websites when the original detector produces $DS_{original} \leq \alpha$.

The autoencoder-based auxiliary model aims to learn the differences between adversarial examples and legitimate websites. As mentioned in the literature review, autoencoders are powerful tools for reconstructing the types of examples they are trained on. Those that cannot be reconstructed well can be classified as a different class. Therefore, our auxiliary model is trained on adversarial examples to accurately reconstruct them. These adversarial examples are generated from the training set of the original detector using the threat model, which transforms detectable phishing websites into adversarial examples. This approach allows the auxiliary model to learn the manipulated patterns derived from the clean data that the original detector has not learned.

Our autoencoder design considers the manipulation actions of adversaries. HTML source code is organized by elements linked in a DOM tree structure. Adversaries may use various evasion techniques to change HTML components, affecting element information (e.g., name, attributes, and text values) and the overall DOM tree structure. Since Graph Convolutional Networks (GCNs) have been shown to effectively represent source code in a syntax tree structure (Ding et al., 2023), we design a GCAE with two GCN layers as the encoder and two GCN layers as the decoder (Figure 2).

The GCAE input consists of an edge matrix and a node matrix derived from the HTML. The edge matrix describes the DOM tree structure, while the node matrix is derived from the node embedding $e_{element}$ which concatenates three components of an element: tag name e_{tag}, attribute names e_{att}, and text e_{text}. Both e_{tag} and e_{att} are extracted via one-hot encoding, and e_{text} is the embedding vector extracted using the pre-trained BERT model. We measure the reconstruction error of our auxiliary model using the mean squared error (MSE) between the input x and the reconstructed output x', denoted as $MSE(x, x')$. We denote the reconstruction error threshold by ε. When websites containing both adversarial examples and true legitimate websites pass through the auxiliary model, the following criterion is used: if a website can be reconstructed with $MSE(x, x') < \varepsilon$, it is likely to be an adversarial example, given that the auxiliary model is trained on adversarial examples. Conversely, if $MSE(x, x') \geq \varepsilon$, it will likely be a legitimate website. In our enhanced detector, the final Decision Score, denoted as DS, is determined through the collaboration of the original detector and the auxiliary model:

If $DS_{original} > \alpha$, then $DS = DS_{original}$.

Else, $DS = \alpha/(1 + \frac{MSE(x,x\prime)}{\varepsilon})$.

The intuition behind this collaborative decision-making process is to use the original detector as the primary model to classify the input as a phishing website when $DS_{original}$ is larger than α. When the $DS_{original}$ falls between 0 and α, the final classification detection considers the GCAE-based auxiliary model as well to distinguish adversarial examples from true legitimate websites. For $DS_{original} \leq \alpha$, DS is calculated by $\alpha/(1 + \frac{MSE(x,x')}{\varepsilon})$. When $MSE(x, x') < \varepsilon$, DS ranges between $\alpha/2$ and α, leading the enhanced detector to classify the input as an adversarial example. Conversely, when $MSE(x, x') \geq \varepsilon$, DS ranges between 0 and $\alpha/2$, leading the enhanced detector to classify the input as a true legitimate website.

5. Evaluation

To evaluate the performance of our adversarial robustness framework, we exhibit the robustness process of an ML-based phishing website detector that is not robust against adversarial attacks. We split 80% of the collected 20,000 clean data into the training set and 20% into the test set. Convolutional Neural Networks (CNN) are popular for designing phishing website detectors, and many adversarial attack studies use CNN-based classifiers as their target detectors (Apruzzese et al., 2022; Montaruli et al., 2023). Therefore, we prepared a CNN-based detector that extracts features from HTML based on the design by Opara et al. (2020). This detector was trained on our training set and will be the target of the threat model. The threshold α for this detector was set as 0.5.

The original detector achieved 88% accuracy on the test set, correctly identifying 1,920 out of 2,000 phishing websites (Table 2, Original). To test the adversarial robustness of the detector, we conducted adversarial attacks using the threat model. It manipulated the 1,920 detectable phishing websites in the test set, transforming them into well-rendered evasive phishing websites as adversarial examples. As the number of manipulations allowed on each detectable phishing website increased, the total number of generated adversarial examples also increased until no more detectable phishing websites could be transformed into evasive cases. In total, 461 out of 1,920 detectable phishing websites were transformed into evasive ones, reducing the detection accuracy from 88% to 76.48% (Figure 3, Panel (A)).

The robustness test demonstrated that the original detector is vulnerable to adversarial attacks,. In our experiment, we compare the performance of three adversarial robustness methods. In addition to our proposed method with the auxiliary model, we have two benchmarks: (1) retraining the original detector from scratch with the training set augmented with adversarial examples and (2) fine-tuning the original detector using adversarial examples. We evaluate performance based on the adversarial robustness to assess the improvement in detecting adversarial attacks and the accuracy of the clean data to determine if enhancing adversarial robustness compromises the high detection performance of the original detector on clean data.

6. Experiments and Results

All three adversarial robustness methods need to be learned from adversarial examples during their training process. Therefore, we use the threat model to generate adversarial examples from the original detector's training set. As a result, 1,928 evasive phishing websites were generated from 7,767 detectable phishing websites in the training set. They represent the unseen phishing patterns that the original detector did not learn.

The retrained detector was trained on an augmented training set with 16,000 clean data points and 1,928 adversarial examples. The fine-tuned detector was fine-tuned on only the 1,928 adversarial examples to adjust the weights of the original detector. The GCAE, used as our auxiliary model, was trained on the 1,928 adversarial examples. We found that the reconstruction error $MSE(x, x')$ for most adversarial examples is smaller than 0.005. Therefore, we set $\varepsilon = 0.005$ as the reconstruction error threshold in the enhanced detector. In Figure 3, panels (B) to (F) show the adversarial robustness tests for the retrained detector, fine-tuned detectors trained for various epochs, and our model.

Table 2. Performance of Detectors on Test Set (clean data).

	Accuracy	Precision	Recall	F1 Score	AUC	TP	FP	FN	TN
(A) Original	***0.88***	***0.89***	***0.88***	***0.8792***	***0.88***	***1920***	***400***	***80***	***1600***
(B) Retrained	0.8972	0.8993	0.8972	0.8971	0.8972	1866	277	134	1723
(C)Fine-tuned (2 epochs)	0.8785	0.891	0.8785	0.8775	0.8785	1936	422	64	1578
(D) Fine-tuned (5 epochs)	0.8678	0.8848	0.8678	0.8663	0.8677	1946	475	54	1525
(E) Fine-tuned (9 epochs)	0.8343	0.8651	0.8343	0.8307	0.8343	1959	622	41	1378
(F) Our Proposed Model	**0.8762**	**0.8872**	**0.8762**	**0.8754**	**0.8763**	**1921**	**416**	**79**	**1584**

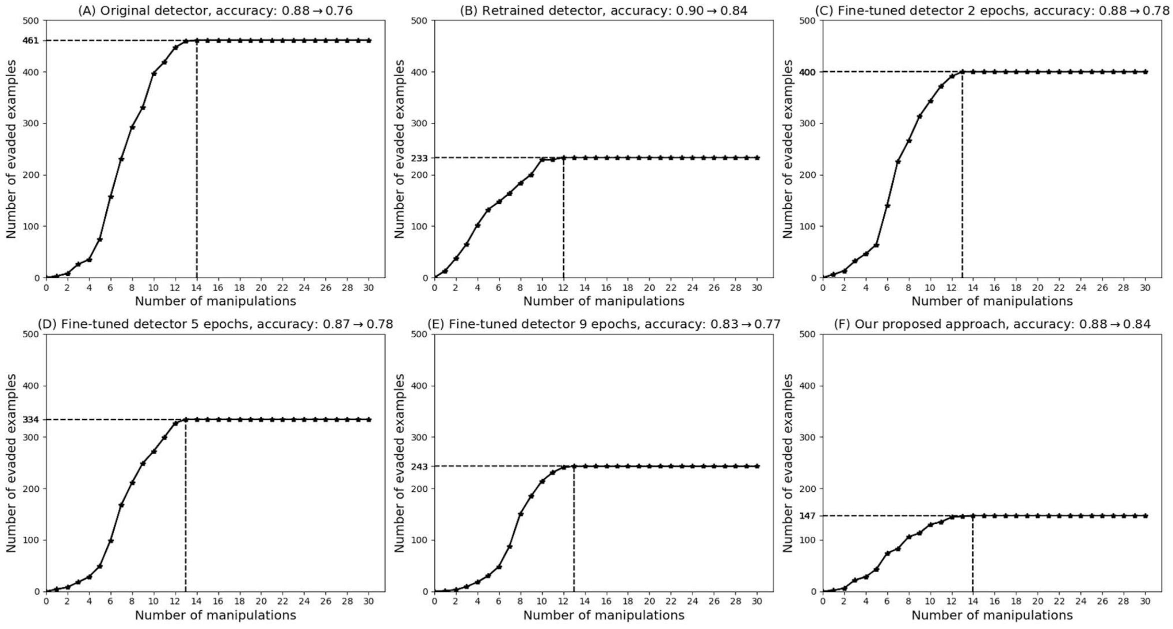

Figure 3. Adversarial Robustness Test of Detectors.

Compared to the original detector (A), which allowed the threat model to generate 461 evasive phishing websites, all adversarial robustness methods enhance the detector's robustness by reducing the number of evasive phishing websites generated. Our proposed model outperforms the others by only allowing 147 detectable phishing websites to be transformed into evasive phishing websites. The overall accuracy of our proposed model after being attacked remains at 84%, which is significantly higher than the original detector. Table 2 compares the performance of detectors on the test set, which reflects their performance on clean data before being attacked.

Our proposed model performs very similarly to the original detector on all nine evaluation metrics (accuracy, precision, recall, F1 score, AUC, TP, FP, FN, and TN). This demonstrates that our proposed model does not compromise the original detector’s high performance on clean data while achieving the highest adversarial robustness among all detectors. This experiment demonstrates the shortcomings of fine-tuning and retraining mentioned in the literature review. The trade-off between clean data performance and adversarial robustness exists for benchmark models. For fine-tuning, we present three versions of the fine-tuned detector trained for 2 epochs, 5 epochs, and 9 epochs. Panels (C) to (E) in Figure 3 and Table 2 show that adversarial robustness increases as the detector is fine-tuned for more epochs; however, clean data performance declines simultaneously. For retraining, the performance on clean data does not decline; rather, it increases slightly. This is because, to achieve the adversarial robustness shown in Figure 3 (B), the model had to be trained for more epochs than the original detector, which improves its performance on clean data. However, its adversarial robustness is still worse than our proposed model as it focuses more on maintaining high performance on clean data.

7. Contributions and Future Directions

Our study contributes two key aspects to the adversarial robustness literature in phishing website detection. First, our adversarial robustness framework emphasizes collaboration between offensive and defensive efforts. Insights gained from the offensive side's realistic emulation of adversarial attacks are applied to enhance ML-based phishing website detectors. Rather than relying on an unrealistic threat model, we improve the robustness of detectors against realistic adversarial attacks by generating well-crafted phishing websites as adversarial examples. Second, to the best of our knowledge, we are among the first to propose an auxiliary model to assist the original detector in filtering out adversarial examples in the context of phishing detection. The collaboration between the auxiliary model and the original detector mitigates the original detector's vulnerability to adversarial examples. As a result, our approach achieves high adversarial robustness while maintaining the original detector's high performance on clean data. Additionally, our framework is not limited to any specific detector; as long as the threat model can manipulate the website components used by the detector, our framework can be employed to test and enhance the detector's adversarial robustness.

The collaborative decision-making between the original detector and the auxiliary model may also be generalizable to other detection domains threatened by evasion problems. For instance, in phishing emails and fake news detection, we can use Large Language Models as auxiliary models to learn the semantic patterns of adversarial examples. In future research, we plan to conduct more experiments to demonstrate the effectiveness of our framework on multiple ML-based phishing website detectors with different designs. We will also introduce our adversarial robustness approach to other domains. Further testing the proposed approach in other contexts can help mitigate the ever-growing issue of fake or fraudulent content generation.

8. References

Abbasi, A., Zahedi, F. “., Zeng, D., Chen, Y., Chen, H., & Nunamaker Jr, J. F. (2015). Enhancing predictive analytics for anti-phishing by exploiting website genre information. *Journal of Management Information Systems*, 31(4), 109–157. https://doi.org/10.1080/07421222.2014.1001260.

Apruzzese, G., Conti, M., & Yuan, Y. (2022). Spacephish: The evasion-space of adversarial attacks against phishing website detectors using machine learning. *Proceedings of the 38th Annual Computer Security Applications Conference*, 171–185. https://doi.org/10.1145/3564625.3567980.

Bai, T., Luo, J., Zhao, J., Wen, B., & Wang, Q. (2021). Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356.* https://doi.org/10.48550/arXiv.2102.01356.

Ding, Z., Li, H., Shang, W. & Chen, T.H. (2023). Towards learning generalizable code embeddings using task-agnostic graph convolutional networks. *ACM Transactions on Software Engineering and Methodology*, 32(2), 1-43. https://doi.org/10.1145/3542944.

Dobolyi, D.G. & Abbasi, A., (2016). Phishmonger: A free and open source public archive of real-world phishing websites. *2016 IEEE conference on intelligence and security informatics (ISI)*, 31-36. https://doi.org/10.1109/ISI.2016.7745439.

Goodfellow, I.J., Shlens, J. & Szegedy, C., 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572.* https://doi.org/10.48550/arXiv.1412.6572.

Hancock, D.Y., Fischer, J., Lowe, J.M., Snapp-Childs, W., Pierce, M., Marru, S., Coulter, J.E., Vaughn, M., Beck, B., Merchant, N. & Skidmore, E. (2021). Jetstream2: Accelerating cloud computing via Jetstream. *In Practice and Experience in Advanced Research Computing*, 1-8. https://doi.org/10.1145/3437359.3465565.

Li, H., Zhou, S., Yuan, W., Luo, X., Gao, C. & Chen, S. (2021). Robust android malware detection against adversarial example attacks. *Proceedings of the Web Conference 2021*, 3603-3612. https://doi.org/10.1145/3442381.3450044.

Ma, J., Saul, L.K., Savage, S. & Voelker, G.M. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URLs. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1245-1254. https://doi.org/10.1145/1557019.1557153.

Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H. & Akoglu, L. (2021). A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12), 12012-12038. https://doi.org/10.1109/TKDE.2021.3118815.

Montaruli, B., Demetrio, L., Pintor, M., Compagna, L., Balzarotti, D. & Biggio, B., 2023, November. Raze to the Ground: Query-Efficient Adversarial HTML Attacks on Machine-Learning Phishing Webpage Detectors. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 233-244. https://doi.org/10.1145/3605764.3623920.

O'Mara, A., Alsmadi, I. & AlEroud, A. (2021). Generative Adverserial Analysis of Phishing Attacks on Static and Dynamic Content of Webpages. *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*, 1657-1662. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00222.

Opara, C., Wei, B. & Chen, Y. (2020). HTMLPhish: enabling phishing web page detection by applying deep learning techniques on HTML analysis. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1-8. https://doi.org/10.1109/IJCNN48605.2020.9207707.

Ouyang, L. & Zhang, Y. (2021). Phishing Web Page Detection with HTML-Level Graph Neural Network. *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 952-958. https://doi.org/10.1109/TrustCom53373.2021.00133.

Pochat, V. L., Van Goethem, T., Tajalizadehkhoob, S., Korczy´nski, M., & Joosen, W. (2019). Tranco: A research-oriented top sites ranking hardened against manipulation. *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2019.23.

Purwanto, R.W., Pal, A., Blair, A. & Jha, S. (2022). Phishsim: Aiding phishing website detection with a feature-free tool. *IEEE Transactions on Information Forensics and Security*, 17, 1497-1512. https://doi.org/10.1109/TIFS.2022.3164212.

Sabir, B., Babar, M.A., Gaire, R. & Abuadbba, A. (2022). Reliability and robustness analysis of machine learning based phishing url detectors. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2022.3218043.

Shirazi, H., Bezawada, B., Ray, I. & Anderson, C. (2021). Directed adversarial sampling attacks on phishing detection. *Journal of Computer Security*, 29(1), 1-23. https://doi.org/10.3233/JCS-191411.

Smadi, S., Aslam, N. & Zhang, L. (2018). Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems*, 107, 88-102. https://doi.org/10.1016/j.dss.2018.01.001.

Song, F., Lei, Y., Chen, S., Fan, L. & Liu, Y. (2021). Advanced evasion attacks and mitigations on practical ML-based phishing website classifiers. *International Journal of Intelligent Systems*, 36(9), 5210-5240. https://doi.org/10.1002/int.22510.

Tian, K., Jan, S.T., Hu, H., Yao, D. & Wang, G. (2018). Needle in a haystack: Tracking down elite phishing domains in the wild. *Proceedings of the Internet Measurement Conference 2018*, 429-442. https://doi.org/10.1145/3278532.3278569.

Wang, H., Chen, T., Gui, S., Hu, T., Liu, J. & Wang, Z. (2020). Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. *Advances in Neural Information Processing Systems*, 33, 7449-7461. https://proceedings.neurips.cc/paper_files/paper/2020/file/537d9b6c927223c796cac288cced29df-Paper.pdf

Xiang, G., Hong, J., Rose, C.P. & Cranor, L. (2011). Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, 14(2), 1-28. https://doi.org/10.1145/2019599.2019606.