applied sciences

IMPACT FACTOR 2.5

CITESCORE 5.3

Article

A Lightweight Multi-View Learning Approach for Phishing Attack Detection Using Transformer with Mixture of Experts

Yanbin Wang, Wenrui Ma, Haitao Xu, Yiwei Liu and Peng Yin

MDPI

https://doi.org/10.3390/app13137429

applied sciences

MDPI

Article

A Lightweight Multi-View Learning Approach for Phishing Attack Detection Using Transformer with Mixture of Experts

Yanbin Wang [1], Wenrui Ma [1], Haitao Xu [1,*], Yiwei Liu [2,*] and Peng Yin [2,3]

[1] School of Cyber and Technology, Zhejiang University, Hangzhou 310027, China; wangyanbin15@mails.ucas.ac.cn (Y.W.); mawenrui@mail.zjgsu.edu.cn (W.M.)
[2] Defence Industry Secrecy Examination and Certification Center, Beijing 100089, China; yinpeng@iie.ac.cn
[3] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 101408, China
* Correspondence: haitaoxu@zju.edu.cn (H.X.); yiweiliu_disecc@163.com (Y.L.)

Abstract: Phishing poses a significant threat to the financial and privacy security of internet users and often serves as the starting point for cyberattacks. Many machine-learning-based methods for detecting phishing websites rely on URL analysis, offering simplicity and efficiency. However, these approaches are not always effective due to the following reasons: (1) highly concealed phishing websites may employ tactics such as masquerading URL addresses to deceive machine learning models, and (2) phishing attackers frequently change their phishing website URLs to evade detection. In this study, we propose a robust, multi-view Transformer model with an expert-mixture mechanism for accurate phishing website detection utilizing website URLs, attributes, content, and behavioral information. Specifically, we first adapted a pretrained language model for URL representation learning by applying adversarial post-training learning in order to extract semantic information from URLs. Next, we captured the attribute, content, and behavioral features of the websites and encoded them as vectors, which, alongside the URL embeddings, constitute the website's multi-view information. Subsequently, we introduced a mixture-of-experts mechanism into the Transformer network to learn knowledge from different views and adaptively fuse information from various views. The proposed method outperforms state-of-the-art approaches in evaluations of real phishing websites, demonstrating greater performance with less label dependency. Furthermore, we show the superior robustness and enhanced adaptability of the proposed method to unseen samples and data drift in more challenging experimental settings.

Keywords: phishing attack detection; multi-view learning; transformer; self-supervised learning

check for updates

Citation: Wang, Y.; Ma, W.; Xu, H.; Liu, Y.; Yin, P. A Lightweight Multi-View Learning Approach for Phishing Attack Detection Using Transformer with Mixture of Experts. *Appl. Sci.* **2023**, *13*, 7429. https://doi.org/10.3390/app13137429

Academic Editor: Mohamed Benbouzid

Received: 31 May 2023
Revised: 14 June 2023
Accepted: 21 June 2023
Published: 22 June 2023

1. Introduce

Phishing attacks have evolved over the years, employing a variety of tactics to deceive users into divulging sensitive information, such as login credentials and financial data [1–6]. The harm caused by phishing attacks is substantial, often resulting in considerable financial losses, identity theft, and damage to the targeted organizations' reputations. In recent years, the number and sophistication of phishing attacks have increased, with cybercriminals continually developing new techniques to evade detection and target unsuspecting victims.

One notable phishing attack in 2016 involved the Bangladesh Bank, where attackers stole USD 81 million by compromising the bank's systems using a phishing email [7]. Another prominent example is the 2020 Twitter Bitcoin scam, where high-profile accounts were hijacked to promote a cryptocurrency scam, causing reputational damage to the social media platform and resulting in financial losses for some users [8]. These incidents highlight the pervasive nature of phishing attacks and the need for effective countermeasures.

Various anti-phishing methods have been proposed to address these threats, ranging from traditional blacklist-based approaches to more sophisticated machine learning techniques. Blacklist-based approaches rely on maintaining a list of known phishing websites, which can quickly become outdated due to the dynamic nature of phishing attacks [9,10]. Machine-learning-based methods offer the potential to automatically learn and adapt to new phishing tactics by analyzing features such as URLs, website content, and behavioral patterns [11]. URL analysis, in particular, has been extensively researched, and advanced representation learning techniques, such as computer vision neural architectures and natural language processing models, have been employed to extract feature patterns from URLs, enabling rapid and accurate phishing URL detection [12–14].

However, these methods have their limitations. For instance, they may struggle to detect attacks that employ advanced obfuscation techniques, such as masquerading URL addresses. Additionally, phishing attackers frequently change their website URLs to evade detection, posing challenges for machine learning models that rely on URL analysis.

Consequently, identifying and leveraging additional features beyond URLs, such as website content, visual appearance, and behavioral patterns, is crucial for building robust anti-phishing countermeasures. Exploring these alternative features can potentially enhance the performance of machine-learning-based phishing detection models. By combining multiple feature sets, it is possible to develop more comprehensive and resilient anti-phishing strategies that can better adapt to the evolving tactics of phishing attackers, ultimately safeguarding users from a wider range of threats.

In this paper, we introduce a lightweight multi-view learning-based approach for detecting phishing websites by considering a diverse and easily obtainable set of website information. Our objective is to provide a highly usable, robust, and accurate solution for phishing detection, aiming to enhance real-world performance, including challenging scenarios such as imbalanced class detection, small-scale datasets used for learning, and long-term model performance. Our contributions are as follows:

- The proposed method employs adversarial post-training to transform a pretrained language model into a powerful URL feature extractor. This approach allows us to transfer the general knowledge learned from massive data by the pretrained language model to URL representation learning at a low cost, resulting in semantic-aware URL embeddings.
- Our research introduces three highly informative website features that are easily obtained yet possess significant discriminative value. These features encompass descriptions of website attributes, content, and behavior, which, in conjunction with the website's URL, form the multi-view information of the website. This allows us to capture phishing attack cues from different perspectives, enabling accurate and evasion-resistant phishing detection.
- We propose utilizing a Transformer network with a mixture-of-experts mechanism to learn the multi-view information of webpages. By dynamically allocating weights to different views and adaptively learning the relevance of features within each view, our method effectively considers the relationships both between and within the constructed views of the webpages.
- The proposed method demonstrates superior performance when evaluated on real phishing websites, outperforming state-of-the-art techniques with higher accuracy and fewer labeled data. Furthermore, it maintains a recognition precision of over 96% even after 3 months, showcasing its robustness and effectiveness in detecting phishing attacks.

2. Related Work

In this section, we review related work in the area of machine-learning-based phishing detection.

2.1. URL-Based Methods

URL-based methods have been widely explored in the literature for phishing website detection. These approaches typically analyze the features and patterns in URLs to identify phishing attempts. Some common URL features include the length of the URL, the presence of special characters, and the use of subdomains. Techniques such as lexical and token analysis can be employed to identify patterns within the URL structure. Researchers have used machine learning algorithms, such as Support Vector Machines (SVMs) and Decision Trees, to classify URLs based on the extracted features [15,16].

Recently, deep learning methods such as convolutional neural networks (CNNs) and natural language processing (NLP) have been applied to automate cybersecurity tasks [17–20], including the automatic extraction of URL features and the detection of phishing websites. Le et al. [21] proposed an end-to-end deep learning framework called URLNet, which uses a convolutional neural network to directly learn non-linear URL embeddings, overcoming the limitations of traditional machine learning methods that only rely on the lexical properties of URL strings and require manual feature engineering. Tajaddodianfar et al. [22] proposed the Texception network, which utilizes character-level and word-level information to predict whether a URL is part of a phishing attack. Jiang et al. [23] proposed an online malicious URL and DNS detection approach based on a character-level deep neural network, which maps URL and DNS strings into vector forms and uses a CNN network framework to automatically extract malicious features and train classification models. Experimental results on real-world URL and DNS datasets demonstrate that this method outperforms several state-of-the-art baseline methods in terms of efficiency and scalability.

Although there are numerous URL-based methods available [24–30], all of these methods may struggle to detect newly created phishing websites or those employing advanced obfuscation techniques, such as masquerading URL addresses. Additionally, phishing attackers frequently change their website URLs to evade detection, posing challenges for URL-based methods.

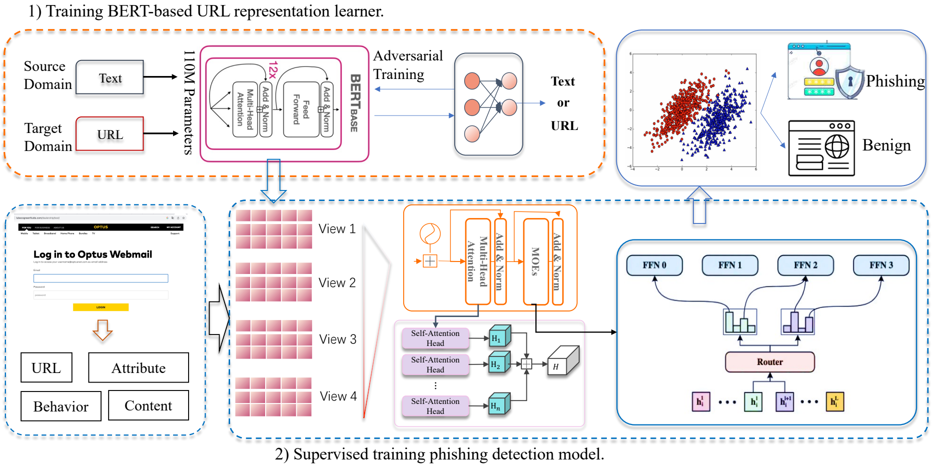
2.2. Methods Leveraging Multi-View Information

Apart from URL-based methods, researchers have also explored alternative approaches that incorporate additional information, such as website content, visual appearance, and behavioral patterns, for phishing detection. Ref. [31] used a hybrid CNN-LSTM model to extract URL features and combined it with page content features to train XGBoost for phishing detection. Ref. [32] proposed a hybrid identity-based phishing detection technique that leverages the webpage's visual and textual identity, incorporating novel image features and brand-specific keywords derived from the webpage content using textual analysis methods. An empirical evaluation on multiple benchmark datasets showed that this joint visual–textual identity detection approach significantly improves the phishing detection performance, with an overall accuracy of 98.6%, while reducing the false-positive rate by up to 3.4%. Ref. [3] used website images and URLs, along with CNNs, to extract features and classify them into benign and phishing pages. HTMLPhish [33] is a language-independent and client-side strategy for webpage phishing detection that utilizes deep-learning-based approaches, specifically CNNs, to learn semantic dependencies in the textual content of HTML webpages and achieved an accuracy and true-positive rate of over 93% on a dataset of more than 50,000 HTML documents. Phish-Sight [34] is a machine-learning-based framework that uses dominant color features and popular brand names embedded in URLs' webpages to detect phishing websites through a visual inspection strategy.

There is also some work using a mixture of features [35–38]. These approaches may require more computational resources and be more complex to implement compared to URL-based methods. Additionally, they may be susceptible to adversarial manipulation of website content or visual appearance. We propose a robust multi-view learning approach that leverages various readily available website information to achieve accurate and resilient phishing detection.

3. Method

The proposed method synergizes several aspects of a website as different views to detect phishing websites, including the website's URL, attribute information, content information, and behavior information. Specifically, we use the following methods: (1) We utilize a small amount of an unlabeled URL corpus and adversarial post-training learning to transform a 12-layer English BERT model into a URL feature extractor. This allows us to obtain the structural and semantic features of URLs at high quality and low cost. (2) Regarding attribute information, we primarily examine the domain registration information of the website. This is based on the consideration that phishing websites often have shorter registration times than benign websites. (3) Content information includes checking pop-ups, overlays, and unusual requests for personal information. (4) The page loading time and redirect times are recorded as the website's behavior information. The statistical results of the aforementioned website's attribute information, content information, and behavioral information are all encoded into vectors of the same scale. Additionally, the features of the URLs are extracted using our well-trained BERT-based encoder. Once all the information from different views has been encoded, we input this information into a Transformer network with a mixture-of-experts mechanism to further learn features relevant to the phishing detection task. This network comprises two key feature weighting components: the multi-head self-attention layer and the mixture-of-experts network. The multi-head self-attention layer captures dependencies between different views and allows the model to simultaneously attend to different semantic information. On the other hand, the mixture-of-experts network combines multiple expert models, which selectively contribute based on the input data's characteristics, thereby enhancing the model's performance and adaptability. The joint utilization of these two components facilitates the effective integration of features from diverse views, enables a comprehensive exploration of diverse information sources, and enables the adaptive extraction of highly expressive features. The model's architectural design is illustrated in the accompanying diagram in Figure 1. To provide a clearer exposition of our method, the contents of this paper are organized as follows: Firstly, we introduce the proposed URL feature learning method. Next, we present the features from perspectives other than the URL, along with their respective feature processing methods. Finally, we describe the multi-view fusion learning architecture.


1) Training BERT-based URL representation learner.

2) Supervised training phishing detection model.

Figure 1. The architecture of the proposed method.

3.1. BERT-Based Universal URL Feature Learning

The advancement of pretrained language models has shown unprecedented progress in data representation. Given the broad similarities between text and URLs, recent efforts have emerged to fine-tune the classical BERT model [39] for learning character representations of URLs, yielding impressive performance in tasks such as phishing and malicious website detection. However, these methods often require a substantial number of labeled URLs for the supervised fine-tuning of BERT, without considering the domain differences between text and URL data during the training process.

To better leverage the knowledge acquired by pretrained models from extensive data, we propose a transfer learning approach based on adversarial domain discrimination learning to adapt the standard classical BERT model for URL representation learning. Adversarial methods typically involve two vital components, namely, the generator and discriminator, with the generator serving as the feature representation learner in this context.

By incorporating adversarial domain discrimination learning, we aim to bridge the domain gap between text and URL data, enabling the BERT model [39] to capture meaningful URL representations without the need for extensive labeled URL data. This approach harnesses the knowledge distilled from pretraining and adapts it to the specific task of URL feature learning, thereby capitalizing on the advantages offered by pretrained models on massive datasets.

The proposed method, inspired by [40], involves two concurrent procedures: post-training and adversarial learning. Specifically, our approach initializes the basic language understanding of BERT [39] with its pretrained weights and adapts BERT through novel self-supervised pretraining tasks, including domain recognition and a target domain masked language model. During the post-training process, we employed adversarial learning to optimize the entire training procedure. The training process is illustrated in Figure 2.

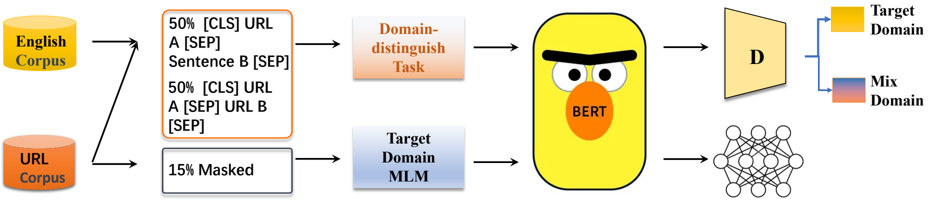


Figure 2. Training process of BERT-based URL feature extractor.

3.1.1. Problem Definition and Notations

We are given two domains, denoted by D_s and D_t, representing the source domain and the target domain, respectively. Within the source domain, there exists a set of unlabeled data D_u^s, where N_u represents the number of unlabeled data points. Similarly, within the target domain, there is another set of unlabeled data D_u^t, with N_t being the number of unlabeled data points. Our objective is to leverage unlabeled data from the target domain to perform unsupervised post-training on the BERT model, enabling it to achieve domain generalization in the target domain (URL) and generate high-quality target representations. This method is named $BERT_{url}$.

3.1.2. Background of BERT

BERT is built on Transformer networks, which employ attention mechanisms to model dependencies between input sequences regardless of their distance. BERT [39] is pretrained using two tasks: masked language modeling (MLM) to predict randomly masked words in the input, and Next Sentence Prediction (NSP) to classify the continuity of sentences. The MLM task allows BERT to capture contextual information from both left and right contexts, while the NSP task enables it to understand relationships between sentences. The pretrained BERT can be easily fine-tuned by adding a softmax output layer for specific classification tasks.

3.1.3. Domain-Distinguish Task

The Next Sentence Prediction (NSP) task encourages BERT to model the relationship between sentences beyond the word level, which proves beneficial for tasks such as question answering and natural language inference. However, in the context of malicious URL detection, we are primarily concerned with analyzing individual URLs, and the inference ability provided by NSP is not required. Instead, the ability to distinguish between different domains plays a crucial role. Therefore, during the post-training procedure, we replace the NSP task with a domain-distinguish task (DDT). Specifically, we construct cross-domain input pairs in the form of [CLS] A [SEP] B [SEP], where [CLS] and [SEP] are special embedding symbols used for classification and sentence separation. In this task, 50% of the time, both A and B are randomly sampled from the target domain data and labeled as TargetDomain. The remaining 50% of the time, A and B are sourced from both the target domain and the source domain, labeled as MixDomain. Domain-distinguish pretraining involves a classification task, where we augment the pooled representation by adding an output layer and maximizing the likelihood of the correct label. By performing domain-distinguish pretraining, we aim to enhance the model's ability to discern distinct domains. This additional layer enables the model to capture specific features relevant to different domains, thereby improving its performance in downstream tasks that involve domain discrimination.

3.1.4. Target Domain MLM

To inject knowledge from the target domain, we employ the masked language model (MLM). This model requires predicting randomly masked words in sentences, encouraging BERT to construct deep bidirectional representations. In our specific task, we assume that there are no labeled data available in the target domain, but there is an abundance of unlabeled data that can be utilized for post-training BERT using MLM.

Specifically, we randomly replace 15% of the words with the [MASK] token. The final hidden vectors corresponding to the masked tokens are then fed into a softmax layer over the vocabulary, maximizing the likelihood of the masked tokens.

Post-training BERT with unlabeled review data from the target domain effectively alleviates the problem of domain knowledge transfer. The overall loss during the post-training process is the sum of the losses from the target domain MLM and the domain-distinguish task.

3.1.5. Adversarial Training

The post-training procedure injects target domain knowledge and endows BERT with domain awareness. Additionally, we employed adversarial training to enhance BERT's perception of domain-specific features in the URL domain. Specifically, we designed a domain discriminator that operates on the hidden state $h_{[CLS]}$ of the special classification embedding [CLS]. This component plays a crucial role in adversarial training, distinguishing the domain characteristics.

The domain discriminator is designed to predict the domain labels of samples, indicating whether they originate from the source or target domain. The parameters of BERT are optimized to maximize the loss of the domain discriminator. This objective encourages BERT to deceive the domain discriminator and generate domain-invariant features.

Specifically, before passing the hidden state $h_{[CLS]}$ of the classification embedding [CLS] to the domain discriminator, it undergoes a gradient reversal layer (GRL) [41]. During forward propagation, the GRL behaves as an identity function. However, during backpropagation, the GRL reverses the gradient by multiplying it with a negative scalar λ. The GRL can be represented as a "pseudo-function" $Q_\lambda(x)$ using the following equations to describe its forward and backward behaviors:

$$Q_\lambda(x) = x \tag{1}$$

$$Q_\lambda(x) = x \tag{2}$$

We denote the hidden state $h_{[CLS]}$ through the GRL by $Q_\lambda\left(h_{[CLS]}\right) = \hat{h}_{[CLS]}$ and then feed it to the domain discriminator as:

$$d = \text{softmax}\left(W_d \hat{h}_{[CLS]} + b_d\right) \tag{3}$$

The objective is to minimize the cross-entropy loss across all data samples originating from both the source and target domains.

$$L_{dom} = -\frac{1}{N_s + N_t} \sum_{i}^{N_s+N_t} \sum_{j}^{K} \hat{d^i}(j) \log d^i(j) \tag{4}$$

where $\hat{d^i} \in 0,1$ denotes the ground-truth domain label. Through the gradient reversal layer (GRL), the parameters θ_d of the domain discriminator are optimized to enhance its predictive capability for domain labels, while the parameters θ_{BERT} of BERT are optimized to deceive the domain discriminator, resulting in the generation of domain-invariant features.

3.2. Additional Views

In addition to the URL, obtaining domain registration information, checking for privacy inquiries and pop-ups, and recording the webpage loading time and redirection counts are essential for constructing different views of a webpage. We employ a WHOIS-based domain query tool [42] to retrieve the registration time of a domain. By calculating the difference between the current date and the domain registration time, we obtain the final domain registration time information. We employ the Selenium tool to validate whether a webpage contains requests for sensitive personal information or exhibits abnormal pop-up windows. If such elements are present, their feature values are set to 1; otherwise, they are set to 0. Additionally, we also collect statistics on the number of webpage redirections and the webpage loading time. To mitigate deviations caused by varying communication conditions, a standardization approach is applied to timestamp webpage loading times. Specifically, we run a set of top 100 websites (representing benign webpages) on a local machine and record their average loading time. Dividing the loading time of each webpage by this average yields the normalized loading time used for subsequent analysis.

Once collected, these pieces of information are encoded into feature vectors with the same dimensionality as the token embeddings of the URL. Specifically, the domain registration information, represented as a numerical value indicating time, is transformed into a binary representation. The resulting binary sequence is then padded with zeros or truncated to create a fixed-length vector of 756 dimensions. Similarly, the content information, redirection count, and webpage loading time are also encoded using the same approach.

3.3. Training with Multi-View Features

After acquiring initial feature representations from multiple views of a website, they are fed into a supervised learning system to train a robust malicious website detection model. During the supervised learning process, our objective is for the model to simultaneously grasp information within each view as well as across views. Learning the information within each view aids in capturing unique characteristics and localized details, leading to a better comprehension of the intrinsic expressions within each view. Simultaneously, learning the information across views enables the model to capture correlations and interdependencies among different viewpoints. This interplay of information allows us to understand the similarities, differences, and complementary nature between views, resulting in a more comprehensive and cohesive feature representation.

To address this, we propose the utilization of a Transformer with a mixture-of-experts (MoE) network to effectively model and integrate multi-view information. This network architecture retains the transformative power of the Transformer's multi-head attention mechanism while replacing the conventional fully connected layers with a flexible process [43]. MoE consists of a gating network and multiple expert networks. The gating network produces a set of gating coefficients that determine the contribution of each expert. These gating coefficients represent the expertise or relevance of each expert for the given input. Each expert network is responsible for learning specific patterns or features in the data. These expert networks are typically designed to capture different aspects of or variations in the data, allowing for a diverse set of representations.

The multi-head attention mechanism empowers the Transformer model to concurrently focus on different positions within the input sequence. By adaptively calculating attention weights, it can effectively capture both the contextual relationships within each view and the associations across views. On the other hand, the mixture-of-experts mechanism, based on a decomposition model, allows the model's decision-making process to be divided into several expert models. These experts' outputs are then combined using weighted combinations. By incorporating the mixture-of-experts mechanism into the Transformer architecture, the model gains the ability to dynamically combine feature representations from diverse views. This synthesis leads to the creation of more expressive and comprehensive multi-view feature representations.

The model architecture is depicted in Figure 1, where the FFN layer is replaced by the MoE layer to facilitate more sophisticated representation learning. Parameter sharing across Transformer blocks enables more efficient parameter deployment. Within each MoE layer, a router is employed to select K experts to acquire more intricate representations.

3.3.1. Computation with MoE

As a representative conditional computation model [44], the MoE model selectively activates a subset of experts within the network. For each input, we specifically direct the relevant hidden representations to the chosen experts for processing. In accordance with the formulation presented by [45], where E denotes the number of trainable experts and $x \in \mathbb{R}^D$ represents the input representation, the output of the MoE model can be expressed as:

$$\text{MoE}(x) = \sum_{i=1}^{E} g(x)_i e_i(x) \tag{5}$$

where $e_i(\cdot)$ denotes a non-linear transformation $\mathbb{R}^D \rightarrow \mathbb{R}^D$ of the ith expert, and $g(\cdot)_i$ is a non-linear mapping $\mathbb{R}^D \rightarrow \mathbb{R}^E$ of the ith element of the output of a trainable router $g(\cdot)$. Usually, both $e(\cdot)$ and $g(\cdot)$ are parameterized by neural networks.

According to the aforementioned formulation, during the training process, $g(\cdot)$ is usually sparsely represented; only a subset of experts can be activated and updated through backpropagation. In this study, each expert is implemented as a Feed-Forward Neural (FFN) layer.

3.3.2. Routing

To enforce sparse routing with $g(\cdot)$, we utilize the TopK() function to select the highest-ranked experts. Following the approach introduced by Riquelme et al. (2021), the formulation of $g(\cdot)$ can be expressed as:

$$g(x) = \text{TopK}(\text{softmax}(f(x) + \epsilon)) \tag{6}$$

Here, $f(\cdot)$ represents the routing linear transformation from $\mathbb{R}^D$ to $\mathbb{R}^E$, and $\epsilon \sim \mathcal{N}\left(0, \frac{1}{E^2}\right)$ is Gaussian noise introduced to facilitate expert routing exploration. The application of softmax after $f(\cdot)$ contributes to enhanced performance and promotes sparsity among the experts. When $K \ll E$, a significant proportion of elements in $g(x)$ tend to approach zero, thereby achieving sparse conditional computation.

4. Experiments

In this section, we empirically evaluate the performance of our proposed methods.

4.1. Implementation Details

During the training of $BERT_{url}$, $BERT_b ase$ (uncased) was utilized as our base model. In the generation of post-training data, each URL was duplicated 10 times with different masks and pairs. To ensure manageable sequence lengths, we set a maximum length limit of 256 tokens. During the post-training phase, we conducted training with a batch size of 16 for a total of 10,000 steps. The optimizer employed was Adam, with a learning rate of 2×10^{-5}, $\beta_1 = 0.8$, $\beta_2 = 0.99$, and an L2 weight decay of 0.01. For the adversarial training, the weights in the domain discriminator were initialized from a truncated normal distribution with a mean of 0.0 and a standard deviation of 0.01. In the gradient reversal layer (GRL), we defined the training progress as $p = \frac{t}{T}$, where t represents the current training step, and T denotes the maximum training step. The adaptation rate λ was dynamically adjusted using the formula $\lambda = \frac{2}{1+\exp(-10p)} - 1$.

We employ a sequence of three consecutive transformer blocks with mixture of experts (MoE) to facilitate the learning of multi-view features. Each block is equipped with eight self-attention heads to capture diverse dependencies within the input. Within the MoE layer, we leverage five experts and set the top K parameter to 2, effectively selecting the most relevant experts for representation fusion. To optimize the model, we utilized the Adam optimizer, configuring the optimizer with a momentum value of 0.9, which promotes smooth and consistent updates, and a weight decay of 0.01 to regulate the magnitude of the model's parameters.

4.2. Dataset

To evaluate the model's performance on phishing URL detection, this study utilized the PhishTank dataset. PhishTank is a publicly available community-driven database of phishing URLs comprising a substantial number of verified phishing websites. The data used for this research were obtained from the PhishTank website (https://www.phishtank.com/). A total of 2,198,157 URLs were collected, predating November 2022, with 3637 online webpages identified as verified phishing webpages.

Given the continuous expansion of the internet, which includes a vast number of URLs, ensuring diversity in the training samples of benign URLs is crucial for training a model so as to accurately reflect the real-world online environment. This diversity ensures that the model is exposed to a wide range of legitimate URLs, helping to mitigate the bias between experimental results and the real world. In order to construct a diverse collection of benign webpages, we sampled a significant number of legitimate URLs from the top 10 million pages based on Open PageRank. Specifically, we initially selected the top 100,000 pages, as they are popular and highly ranked, making them less likely to be disguised as phishing websites. Subsequently, we randomly sampled from the remaining 9.9 million pages to ensure a diverse distribution of the collected benign webpages, approximating the overall distribution of the internet. Finally, we verified the sampled webpages using the VirusTotal tool to ensure their legitimacy.

Consequently, a curated set of 40,066 benign webpages was obtained. An unlabeled URL dataset was created using the 2,198,157 URLs combined with 40,066 benign URLs, which was utilized for the adversarial pretraining of $BERT_{url}$. For the training and evaluation of the final phishing detection model, 3637 verified phishing webpages along with 40,066 benign webpages were employed.

This meticulous data collection process, encompassing a comprehensive range of both phishing and benign URLs, ensures the robustness and reliability of the model's performance assessment in the phishing URL detection domain.

4.3. Baseline Approaches

In this study, several state-of-the-art techniques were selected as baseline approaches and compared with the proposed method. These methods include URLNet [21], DURLD [46], DTD [47], FS-NB [48], TCURL [49], and PhishBERT [27].

URLNet is a combination of character-level and word-level convolutional neural networks, designed to capture multi-scale URL features and improve phishing URL detection. DURLD encodes the original URL into character-level embeddings and leverages the state-of-the-art Transformer architecture to train the malicious detection model. TCURL adopts a hybrid network structure with two parallel branches: a convolutional branch and a Transformer branch. A fusion block is used to process information from both branches, considering both local and global correlations of URL characters. FS-NB is a supervised feature selection method based on statistical approaches, aiming to enhance the performance of

URL-based webpage topic classifiers. This method incorporates embedded feature selection to achieve feature-weighted Naive Bayes classification. Lastly, DTD is a deep-learning-based method for web tracking. PhishBERT, a dedicated URL pretraining language model, employs the standard BERT model architecture and leverages masked language modeling along with a customized pretraining objective. It undergoes pretraining on a vast dataset of over 2 billion URL samples.

4.4. Experimental Setting

To evaluate the performance of the model, we devised three experimental scenarios: (1) balanced dataset evaluation, where the model is trained using an equal number of positive and negative samples; (2) imbalanced dataset evaluation, which represents a more realistic setting mimicking the actual internet environment; (3) data drift evaluation, in which, due to the constant efforts of phishing attackers to evade detection by frequently updating or evolving webpages, the occurrence of concept drift becomes inevitable. Consequently, even well-trained models may experience a decline in their detection performance over a certain period. In this experimental setting, we assessed the model's sustained detection capability.

4.5. Evaluation Metrics

In this study, we employed the following evaluation metrics to assess the proposed methods:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$TPR = \frac{TP}{TP + FN} \tag{8}$$

$$FPR = \frac{FP}{TN + FP} \tag{9}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

In addition to these measures, we also report the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) as part of our findings.

4.6. Evaluation on Balanced Dataset

In this experiment, a balanced dataset was created by merging 3637 online phishing webpages with an equal number of randomly sampled webpages from a pool of 40,066 benign webpages. The average results of several methods, obtained through five-fold cross-validation, are presented in Table 1.

Table 1. Comparison of results: our method vs. baseline method on balanced dataset using five-fold cross-validation.

	URLNet	**DURLD**	**DTD**	**FS-NB**	**TCURL**	**PhishBERT**	**Our Method**
Accuracy	0.9793	0.9704	0.9767	0.9718	0.9748	0.9856	0.9861
TPR	0.9839	0.9795	0.9698	0.9681	0.9714	0.9852	0.9896
FPR	0.0253	0.0387	0.0164	0.0245	0.0218	0.0140	0.0174
AUC	0.9843	0.9813	0.9751	0.9736	0.9787	0.9905	0.9932
F1-Score	0.9794	0.9707	0.9765	0.9717	0.9747	0.9856	0.9861

Compared to other methods, our approach demonstrates significant advantages across multiple evaluation metrics, achieving an accuracy of 0.9861 and a high TPR with a low FPR.

Additionally, our method demonstrates remarkable performance in two comprehensive metrics, F1-score and AUC, with values of 0.9222 and 0.9932, respectively. These results indicate the advantages of our approach in simultaneously maintaining high precision and recall rates, as well as its strong discriminative power and robustness in phishing/benign sample differentiation.

4.7. Evaluation on Unbalanced Dataset

The quantity of phishing webpages on the internet is typically much lower compared to that of legitimate webpages. Consequently, it is paramount to employ an imbalanced dataset for model evaluation. We curated an imbalanced dataset consisting of 3637 verified phishing webpages and 40,066 benign webpages. The benign webpages were sampled from the top 10 million websites, providing a larger sampling space than any previous experimental setup to ensure the diversity of benign samples.

Therefore, our experimental setup presents two key challenges compared to prior research: (1) We deliberately introduced a significantly large class imbalance, where the model tends to focus more on the plentiful benign webpage samples, potentially resulting in a diminished capability to accurately predict positive samples. (2) Due to the diversity and comprehensiveness of our benign samples, the test set contains negative sample patterns and features that are unseen or infrequently encountered in the training set. This poses a risk of the model misclassifying samples that should actually belong to the negative class as positive, leading to an increase in false positives.

Table 2 presents the results of these models using different training data sizes (70%, 30%, and 10% of the total dataset).

Table 2. Performance comparison of proposed method and baseline methods on imbalanced dataset with varying training set sizes.

		URLNet	**DURLD**	**DTD**	**FS-NB**	**TCURL**	**PhishBERT**	**Our Method**
70% of Data	Accuracy	0.9756	0.9752	0.9661	0.9679	0.9737	0.9810	0.9853
	TPR	0.9698	0.9620	0.9558	0.9565	0.9619	0.9730	0.9839
	FPR	0.0239	0.0236	0.0330	0.0311	0.0252	0.0183	0.0146
	F1-Score	0.8687	0.8659	0.8244	0.8322	0.8589	0.8950	0.9176
30% of Data	Accuracy	0.9175	0.9084	0.8962	0.9063	0.8918	0.9237	0.9567
	TPR	0.8908	0.8809	0.8596	0.8870	0.8632	0.8991	0.9395
	FPR	0.0801	0.0891	0.1005	0.0919	0.1056	0.0741	0.0417
	F1-Score	0.6425	0.6155	0.5796	0.6118	0.5704	0.6623	0.7832
10% of Data	Accuracy	0.7887	0.7927	0.7673	0.7438	0.8080	0.8458	0.8878
	TPR	0.7320	0.7348	0.7059	0.7060	0.7658	0.8000	0.8708
	FPR	0.2062	0.2020	0.2271	0.2528	0.1882	0.1500	0.1107
	F1-Score	0.3657	0.3711	0.3355	0.3145	0.3990	0.4634	0.5637

When utilizing 70% of the data for training, our method achieved an accuracy of 0.9853, outperforming all the other methods, including URLNet, DURLD, DTD, FS-NB, TCURL, and PhishBERT, which achieved accuracies ranging from 0.9679 to 0.9810. Additionally, our method demonstrated a high true-positive rate (TPR) of 0.9839, indicating its ability to effectively identify phishing webpages. Moreover, our method exhibited a lower false-positive rate (FPR) of 0.0146, suggesting its capacity to minimize the misclassification of benign webpages as phishing. The F1-score, a comprehensive measure of precision and recall, was significantly higher for our method at 0.9176, while the baseline methods ranged from 0.8244 to 0.8950.

When training with only 30% of the data, our method continued to outperform the baselines, achieving an accuracy of 0.9567. Notably, our method exhibited a higher TPR of 0.9395 compared to the other methods, indicating its robustness in detecting phishing webpages. The FPR for our method was notably low at 0.0417, showcasing its ability to maintain a low misclassification rate of benign webpages. The F1-score for our method was 0.7832, surpassing those of the baseline methods, which ranged from 0.5704 to 0.6623.

Even with a limited training set size of 10%, our method demonstrated commendable performance, achieving an accuracy of 0.8878. It showcased a relatively high TPR of 0.8708, implying its capability to identify a considerable proportion of phishing webpages. Furthermore, our method achieved a low FPR of 0.1107, indicating its ability to effectively distinguish benign webpages. The F1-score for our method was 0.5637, surpassing those of the baseline methods, which ranged from 0.3145 to 0.4634.

Overall, our method consistently outperformed the baseline methods across varying training set sizes. It showcased robustness in identifying phishing webpages, maintaining a low misclassification rate of benign webpages and achieving higher F1-scores. These results highlight the effectiveness and superiority of our method in tackling the challenges posed by imbalanced datasets and varying training set sizes.

Furthermore, we plotted the ROC curves and calculated the corresponding AUC values for several models under this experimental setup.

Figure 3 demonstrates that the proposed method exhibits higher confidence in distinguishing benign webpages from phishing webpages while achieving a favorable balance between sensitivity and specificity. It effectively maintains a high true-positive rate while reducing the false-positive rate. Moreover, the superior ROC curve and higher AUC value indicate the model's robustness, enabling it to handle imbalanced datasets. It remains unaffected by the uneven distribution of positive and negative samples, thereby adapting well to diverse data scenarios and providing relatively stable evaluations.

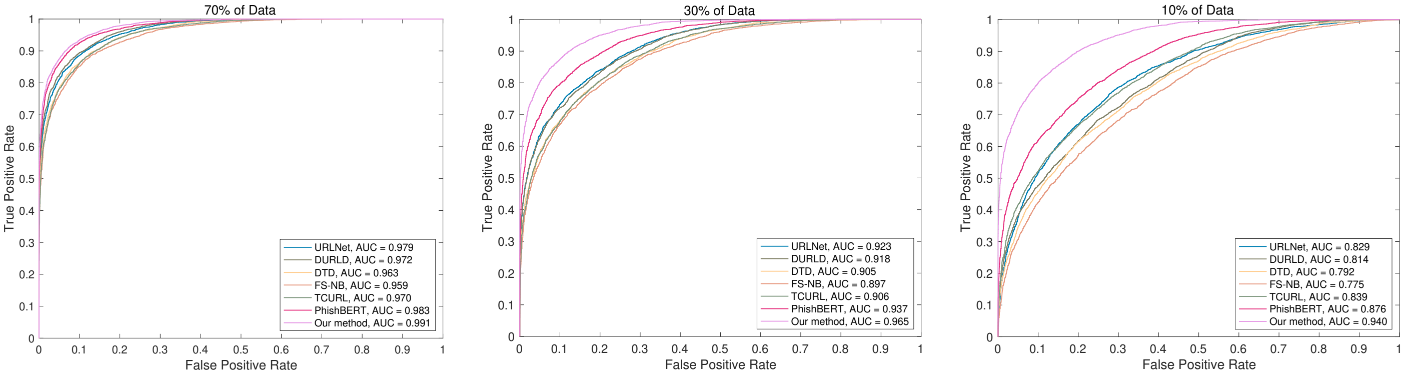
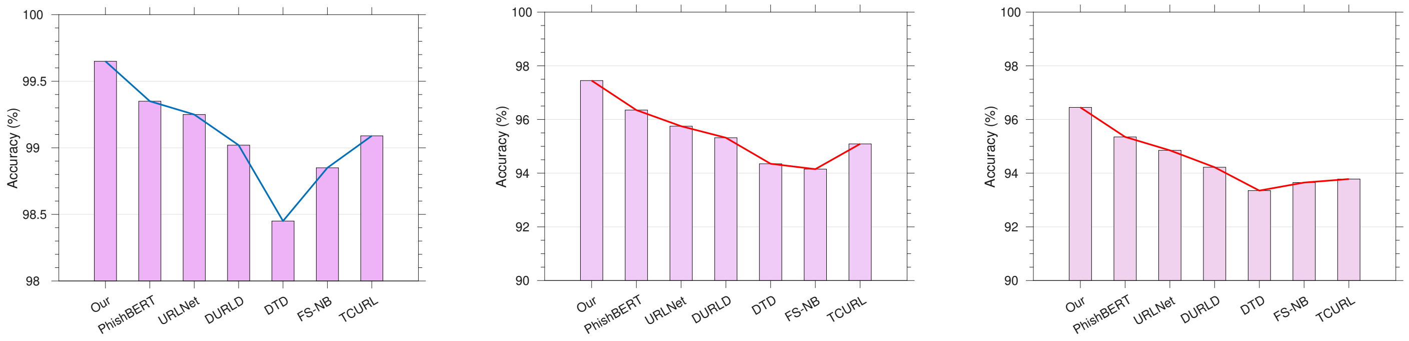


Figure 3. Comparison of ROC and AUC between proposed method and baseline method in imbalanced data experiment.

4.8. Data Drift Evaluation

Phishers continuously modify or migrate the URLs of their phishing webpages to evade security mechanisms and countermeasures, enabling them to deceive users and engage in malicious activities. This strategy makes the detection and prevention of phishing webpages increasingly challenging, as the dynamic changes in webpage addresses render learned rules and patterns from the training set ineffective. This phenomenon is commonly referred to as data drift. To evaluate the model's resilience to data drift, we collected an evaluation dataset over a period of 3 months after completing the training data collection. Each month, we continued to gather 100 samples, forming the data drift evaluation dataset consisting of three subsets, C1 to C3, representing phishing webpages collected one month, two months, and three months after model training, respectively. Since the evaluation dataset solely consists of phishing samples, we report only the accuracy.

The results in Figure 4 indicate that our proposed method outperforms all six baseline approaches in handling data drift in phishing websites. Upon analyzing the results in detail, we observe several key factors that contribute to the superior performance of our method. Firstly, our approach incorporates advanced feature engineering techniques and learning algorithms that capture both static and dynamic characteristics of phishing websites. By considering features such as URL structure and behavioral patterns, our method can effectively identify subtle changes introduced by data drift.



(**a**) C1 (**b**) C2 (**c**) C3

Figure 4. Comparison between proposed method and baseline method in data drift experiment.

4.9. Ablation Study

We conducted a series of experiments comparing the performance of our method with and without key components:

1. $BERT_{url}$: This scheme removes perspectives other than the URL and utilizes only our adversarially trained BERTurl model.
2. $BERT + Multi\text{-}views$: This scheme utilizes the original English BERT to obtain the initial representation of the URL and combines it with several other perspective features used in this study.
3. $FinedBERT + Multi\text{-}views$: This scheme utilizes the fine-tuned BERT for the URL, in conjunction with other perspective features used in this study.
4. $Full$.

The ablation study was conducted on the imbalanced dataset. By analyzing the ROC curves and corresponding AUC values, we can assess the relative effectiveness of each ablation model.

Based on Figure 5, we observed several key findings regarding the performance of different models. Firstly, the Full model consistently outperformed all the ablation models across various evaluation metrics. Notably, as the amount of training data decreased, the performance gap between the Full model and the ablation models widened. This finding highlights the importance of retaining all the components and features in the Full model, as it consistently demonstrated superior performance in handling the complexities of the task.

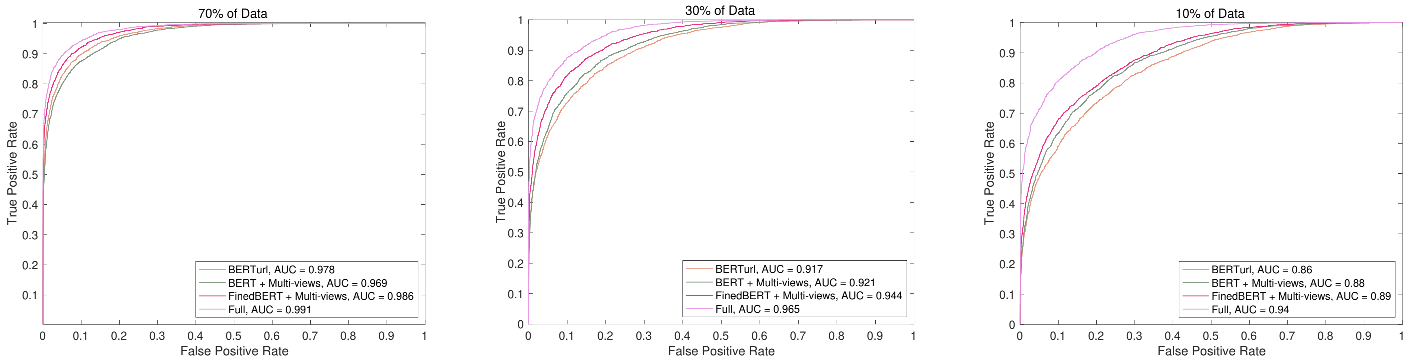


Figure 5. Comparison of several ablation models.

Secondly, we compared the performance of Ablation Model 1 ($BERT_{url}$) and Ablation Model 2 ($BERT + Multi\text{-}views$) across three different datasets. Interestingly, we observed that as the dataset size decreased, the importance of the multi-perspective information increased. Specifically, the performance of Ablation Model 1, which retained the smile feature, was better than Ablation Model 2 on smaller datasets. This suggests that the inclusion of multiple perspectives becomes more critical when the dataset is limited, as it provides additional discriminative information for accurate classification.

Overall, these results provide valuable insights into the effectiveness of the Full model and highlight the significance of multiple perspectives in capturing the complexities of the task, particularly in scenarios with limited training data. These findings contribute to the understanding and improvement of models for the task at hand, with implications for further research and development in related fields.

5. Discussion

Here, we discuss the competitiveness and limitations of the proposed method compared to existing state-of-the-art approaches, as well as scalability and deployment considerations.

5.1. Competitive Analysis

In addition to outperforming other state-of-the-art methods in baseline experimental comparisons, our approach demonstrates significantly superior performance in areas such as imbalanced data learning, small-scale data learning, and long-term model performance. The outstanding performance of our proposed model can be attributed to several key factors:

- Our model incorporates multiple perspectives by extracting various features from a webpage, including its URL, domain registration, privacy requests, loading time, and redirection information, which are closely related to phishing websites. By considering these features, our model is able to capture various aspects and complex patterns associated with phishing attempts, thereby improving performance and enhancing the handling of small sample learning and class imbalance scenarios.
- Furthermore, we utilize a Transformer network with mixture of experts to integrate multi-perspective features for supervised learning in phishing detection tasks. The combination of Transformer's attention mechanism and mixture of experts allows the model to extract essential knowledge from multi-perspective information, accurately learn class distribution boundaries in class-imbalanced data, and particularly focus on highly expressive and stable features, thereby enhancing its long-term detection capability.
- URLNet [21], DURLD [46], and PhishBERT [27] have demonstrated the importance of utilizing URLs. However, these methods are constrained by the limited scope of labeled learning. To address this limitation, we propose an unsupervised URL feature learning method based on BERT. This method incorporates adversarial and domain-aware learning, taking into consideration resource efficiency and leveraging the powerful representation capabilities of pretrained models. Our approach not only enables us to learn the structural and semantic information of URLs but also allows us to acquire expanded knowledge from models trained on massive amounts of text. These contributions serve as a solid foundation for the performance of our proposed model.

5.2. Analysis of Limitations and Scalability

5.2.1. Scalability

Our approach is built entirely upon the Transformer architecture, which provides better scalability compared to previous methods based on CNNs, RNNs, and hybrid networks. Transformer enables efficient parallel processing and distributed computing, allowing it to handle large-scale datasets and adapt to increasing workloads. Additionally, the modular and hierarchical structure of Transformer allows for easy scalability and the integration of additional components, improving the overall system's scalability.

5.2.2. Limitations

Our lightweight method for multi-view feature learning faces challenges when deployed on edge devices due to the use of a BERT-based approach for URL representation learning. However, we can overcome these challenges through efficient compression techniques and model optimization methods.

To address deployment difficulties on edge devices, we can leverage techniques such as model quantization, knowledge distillation, and parameter pruning. Model quantization reduces the network weight and activation precision, minimizing memory usage and computational requirements. Knowledge distillation transfers knowledge from a large model to a smaller, efficient model while maintaining performance. Parameter pruning selectively removes redundant or less important parameters, further reducing size and computational demands.

By applying these techniques, we strike a balance between model complexity and efficiency, facilitating deployment on edge devices with limited computational resources.

6. Conclusions

In conclusion, this study addresses the significant threat posed by phishing attacks to internet users' financial and privacy security. We propose a robust, multi-view Transformer model with an expert-mixture mechanism for the accurate detection of phishing websites. By incorporating URL analysis, website attributes, content, and behavioral information, our approach surpasses existing methods in terms of performance and label dependency. Specifically, we employed adversarial post-training to adapt a pretrained language model for URL representation learning, enabling the extraction of semantic information from URLs. Furthermore, we captured the attribute, content, and behavioral features of the websites and encoded them as vectors. These, along with the URL embeddings, form the multi-view information of the websites. Our study contributes to the advancement of phishing detection methods by introducing a comprehensive and robust approach that leverages multiple views and exhibits high performance even in challenging scenarios. The proposed method holds promise for enhancing the security of internet users and countering the evolving tactics employed by phishing attackers.

Author Contributions: Conceptualization, Y.W. and H.X.; methodology, Y.L.; software, P.Y.; validation, W.M.; resources and data curation, Y.W.; writing—original draft preparation. All authors have read and agreed to the published version of the manuscript.

Funding: We gratefully acknowledge the financial support received from the following funding sources to conduct this research. The research was partially supported by the National Natural Science Foundation of China under Grant No. 62272410, the Defense Industrial Technology Development Program under Grant JCKY2021906B002 and Grant JCKY2021602B002.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be found on PhishTank.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

URL	Uniform Resource Locator
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve

References

1. Zabihimayvan, M.; Doran, D. Fuzzy rough set feature selection to enhance phishing attack detection. In Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 23–26 June 2019; pp. 1–6.

2. Basnet, R.; Mukkamala, S.; Sung, A.H. Detection of phishing attacks: A machine learning approach. *Soft Comput. Appl. Ind.* **2008**, *226*, 373–383.
3. Al-Ahmadi, S. A deep learning technique for web phishing detection combined URL features and visual similarity. *Int. J. Comput. Netw. Commun. (IJCNC)* **2020**, *12*, 41–54. [CrossRef]
4. Cui, Q.; Jourdan, G.V.; Bochmann, G.V.; Couturier, R.; Onut, I.V. Tracking phishing attacks over time. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 667–676.
5. Goel, D.; Jain, A.K. Mobile phishing attacks and defence mechanisms: State of art and open research challenges. *Comput. Secur.* **2018**, *73*, 519–544. [CrossRef]
6. Prakash, P.; Kumar, M.; Kompella, R.R.; Gupta, M. Phishnet: Predictive blacklisting to detect phishing attacks. In Proceedings of the 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010; pp. 1–5.
7. Sarker, B.; Sarker, B.; Podder, P.; Robel, M. Progression of Internet Banking System in Bangladesh and its Challenges. *Int. J. Comput. Appl.* **2020**, *177*, 11–15. [CrossRef]
8. Okereafor, K.; Adelaiye, O. Randomized cyber attack simulation model: A cybersecurity mitigation proposal for post covid-19 digital era. *Int. J. Recent Eng. Res. Dev. (IJRERD)* **2020**, *5*, 61–72.
9. Moghimi, M.; Varjani, A.Y. New rule-based phishing detection method. *Expert Syst. Appl.* **2016**, *53*, 231–242. [CrossRef]
10. Adewole, K.S.; Akintola, A.G.; Salihu, S.A.; Faruk, N.; Jimoh, R.G. Hybrid rule-based model for phishing URLs detection. In *Emerging Technologies in Computing, Proceedings of the Second International Conference, iCETiC 2019, London, UK, 19–20 August 2019*; Proceedings 2; Springer: Cham, Switzerland, 2019; pp. 119–135.
11. Blum, A.; Wardman, B.; Solorio, T.; Warner, G. Lexical feature based phishing URL detection using online learning. In Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, Chicago, IL, USA, 8 October 2010; pp. 54–60.
12. Saxe, J.; Berlin, K. eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. *arXiv* **2017**, arXiv:1702.08568.
13. Afroz, S.; Greenstadt, R. Phishzoo: Detecting phishing websites by looking at them. In Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing, Palo Alto, CA, USA, 18–21 September 2011; pp. 368–375.
14. Liu, R.; Lin, Y.; Yang, X.; Ng, S.H.; Divakaran, D.M.; Dong, J.S. Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 1633–1650.
15. Mahajan, R.; Siddavatam, I. Phishing website detection using machine learning algorithms. *Int. J. Comput. Appl.* **2018**, *181*, 45–47. [CrossRef]
16. Ahammad, S.H.; Kale, S.D.; Upadhye, G.D.; Pande, S.D.; Babu, E.V.; Dhumane, A.V.; Bahadur, M.D.K.J. Phishing URL detection using machine learning methods. *Adv. Eng. Softw.* **2022**, *173*, 103288. [CrossRef]
17. Heidari, A.; Jamali, M.A.J.; Navimipour, N.J.; Akbarpour, S. A QoS-Aware Technique for Computation Offloading in IoT-Edge Platforms Using a Convolutional Neural Network and Markov Decision Process. *IT Prof.* **2023**, *25*, 24–39. [CrossRef]
18. Heidari, A.; Navimipour, N.J.; Unal, M. A Secure Intrusion Detection Platform Using Blockchain and Radial Basis Function Neural Networks for Internet of Drones. *IEEE Internet Things J.* **2023**, *10*, 8445–8454. [CrossRef]
19. Catillo, M.; Pecchia, A.; Villano, U. A Deep Learning Method for Lightweight and Cross-Device IoT Botnet Detection. *Appl. Sci.* **2023**, *13*, 837. [CrossRef]
20. Nwakanma, C.I.; Ahakonye, L.A.C.; Njoku, J.N.; Odirichukwu, J.C.; Okolie, S.A.; Uzondu, C.; Ndubuisi Nweke, C.C.; Kim, D.S. Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review. *Appl. Sci.* **2023**, *13*, 1252. [CrossRef]
21. Le, H.; Pham, Q.; Sahoo, D.; Hoi, S.C. URLNet: Learning a URL representation with deep learning for malicious URL detection. *arXiv* **2018**, arXiv:1802.03162.
22. Tajaddodianfar, F.; Stokes, J.W.; Gururajan, A. Texception: A character/word-level deep learning model for phishing URL detection. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2857–2861.
23. Jiang, J.; Chen, J.; Choo, K.K.R.; Liu, C.; Liu, K.; Yu, M.; Wang, Y. A deep learning based online malicious URL and DNS detection scheme. In *Security and Privacy in Communication Networks, Proceedings of the 13th International Conference, Secure Comm 2017, Niagara Falls, ON, Canada, 22–25 October 2017*; Proceedings 13; Springer: Cham, Switzerland, 2018; pp. 438–448.
24. Alshehri, M.; Abugabah, A.; Algarni, A.; Almotairi, S. Character-level word encoding deep learning model for combating cyber threats in phishing URL detection. *Comput. Electr. Eng.* **2022**, *100*, 107868. [CrossRef]
25. Aljabri, M.; Mirza, S. Phishing attacks detection using machine learning and deep learning models. In Proceedings of the 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 1–3 March 2022; pp. 175–180.
26. Patgiri, R.; Biswas, A.; Nayak, S. deepBF: Malicious URL detection using learned bloom filter and evolutionary deep learning. *Comput. Commun.* **2023**, *200*, 30–41. [CrossRef]
27. Wang, Y.; Zhu, W.; Xu, H.; Qin, Z.; Ren, K.; Ma, W. A Large-Scale Pretrained Deep Model for Phishing URL Detection. In Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.

28. Xuan, C.D.; Nguyen, H.D.; Tisenko, V.N. Malicious URL detection based on machine learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 148–153. [CrossRef]
29. Wu, T.; Wang, M.; Xi, Y.; Zhao, Z. Malicious URL Detection Model Based on Bidirectional Gated Recurrent Unit and Attention Mechanism. *Appl. Sci.* **2022**, *12*, 12367. [CrossRef]
30. Abdul Samad, S.R.; Balasubaramanian, S.; Al-Kaabi, A.S.; Sharma, B.; Chowdhury, S.; Mehbodniya, A.; Webber, J.L.; Bostani, A. Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection. *Electronics* **2023**, *12*, 1642. [CrossRef]
31. Ozcan, A.; Catal, C.; Donmez, E.; Senturk, B. A hybrid DNN–LSTM model for detecting phishing URLs. *Neural Comput. Appl.* **2023**, *35*, 4957–4973. [CrossRef]
32. Tan, C.C.L.; Chiew, K.L.; Yong, K.S.; Sebastian, Y.; Than, J.C.M.; Tiong, W.K. Hybrid phishing detection using joint visual and textual identity. *Expert Syst. Appl.* **2023**, *220*, 119723. [CrossRef]
33. Opara, C.; Wei, B.; Chen, Y. HTMLPhish: Enabling phishing web page detection by applying deep learning techniques on HTML analysis. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
34. Pandey, P.; Mishra, N. Phish-Sight: A new approach for phishing detection using dominant colors on web pages and machine learning. *Int. J. Inf. Secur.* **2023**, 1–11. [CrossRef]
35. Aljofey, A.; Jiang, Q.; Rasool, A.; Chen, H.; Liu, W.; Qu, Q.; Wang, Y. An effective detection approach for phishing websites using URL and HTML features. *Sci. Rep.* **2022**, *12*, 8842. [CrossRef] [PubMed]
36. Benavides-Astudillo, E.; Fuertes, W.; Sanchez-Gordon, S.; Rodriguez-Galan, G.; Martínez-Cepeda, V.; Nuñez-Agurto, D. Comparative Study of Deep Learning Algorithms in the Detection of Phishing Attacks Based on HTML and Text Obtained from Web Pages. In *International Conference on Applied Technologies, Proceedings of the 4th International Conference, ICAT 2022, Quito, Ecuador, 23–25 November 2022*; Springer: Cham, Switzerland, 2022; pp. 386–398.
37. Paturi, R.; Swathi, L.; Pavithra, K.S.; Mounika, R.; Alekhya, C. Detection of Phishing Attacks using Visual Similarity Model. In Proceedings of the 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 9–11 May 2022; pp. 1355–1361.
38. Ariyadasa, S.; Fernando, S.; Fernando, S. Combining Long-Term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites Using URL and HTML. *IEEE Access* **2022**, *10*, 82355–82375. [CrossRef]
39. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
40. Du, C.; Sun, H.; Wang, J.; Qi, Q.; Liao, J. Adversarial and domain-aware BERT for cross-domain sentiment analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 June 2020; pp. 4019–4028.
41. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial discriminative domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7167–7176.
42. Shi, Y.; Chen, G.; Li, J. Malicious domain name detection based on extreme machine learning. *Neural Process. Lett.* **2018**, *48*, 1347–1357. [CrossRef]
43. Xue, F.; Shi, Z.; Wei, F.; Lou, Y.; Liu, Y.; You, Y. Go wider instead of deeper. *AAAI Conf. Artif. Intell.* **2022**, *36*, 8779–8787. [CrossRef]
44. Bengio, Y. Deep learning of representations: Looking forward. In *Statistical Language and Speech Processing, Proceedings of the First International Conference, SLSP 2013, Tarragona, Spain, 29–31 July 2013*; Proceedings 1; Springer: Cham, Switzerland, 2013; pp. 1–37.
45. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv* **2017**, arXiv:1701.06538.
46. Srinivasan, S.; Vinayakumar, R.; Arunachalam, A.; Alazab, M.; Soman, K. DURLD: Malicious URL detection using deep learning-based character level representations. In *Malware Analysis Using Artificial Intelligence and Deep Learning*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 535–554.
47. Castell-Uroz, I.; Poissonnier, T.; Manneback, P.; Barlet-Ros, P. URL-based Web tracking detection using deep learning. In Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2–6 November 2020; pp. 1–5.
48. Rajalakshmi, R.; Aravindan, C. A Naive Bayes approach for URL classification with supervised feature selection and rejection framework. *Comput. Intell.* **2018**, *34*, 363–396. [CrossRef]
49. Wang, C.; Chen, Y. TCURL: Exploring hybrid transformer and convolutional neural network on phishing URL detection. *Knowl.-Based Syst.* **2022**, *258*, 109955. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.