electronics

MDPI

Article

Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing URL Detection

Seok-Jun Bu and Sung-Bae Cho *

Department of Computer Science, Yonsei University, Seoul 03722, Korea; sjbuhan@yonsei.ac.kr
* Correspondence: sbcho@yonsei.ac.kr

Abstract: Considering the fatality of phishing attacks, the data-driven approach using massive URL observations has been verified, especially in the field of cyber security. On the other hand, the supervised learning approach relying on known attacks has limitations in terms of robustness against zero-day phishing attacks. Moreover, it is known that it is critical for the phishing detection task to fully exploit the sequential features from the URL characters. Taken together, to ensure both sustainability and intelligibility, we propose the combination of a convolution operation to model the character-level URL features and a deep convolutional autoencoder (CAE) to consider the nature of zero-day attacks. Extensive experiments on three real-world datasets consisting of 222,541 URLs showed the highest performance among the latest deep-learning methods. We demonstrated the superiority of the proposed method by receiver-operating characteristic (ROC) curve analysis in addition to 10-fold cross-validation and confirmed that the sensitivity improved by 3.98% compared to the latest deep model.

Keywords: phishing detection; zero-day attack; anomaly detection; convolutional autoencoder; deep learning

check for updates

Citation: Bu, S.-J.; Cho, S.-B. Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing URL Detection. *Electronics* **2021**, *10*, 1492. https://doi.org/10.3390/electronics10121492

Academic Editor: Younho Lee

Received: 1 June 2021
Accepted: 18 June 2021
Published: 21 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A phishing attack in its broadest sense can be defined as a scalable act of deception whereby impersonation is used by an attacker to obtain information from an individual [1]. Considering that the most common form of online phishing attacks is malicious hyperlinks embedded in messages, the recent technological trend in which personal connections are reinforced due to the explosive growth of social media services is particularly vulnerable. Consequently, it is important to conduct a study on better understanding the diffusion of phishing URLs for improving the safety and reliability of devices and networks [2].

In the field of cyber security, the supervised learning approach to learn the features from phishing attacks based on various machine-learning techniques with massive known-attack observations was introduced [3,4]. Deep learning is a representative method of learning the mapping function between observed URL features and labels through a large number of parameters (weights) expressed by the layer-by-layer matrix product and sum operation. Among the most prominent methods, the combination of a convolutional neural network (CNN) and a recurrent neural network (RNN) has been found to significantly improve the detection performance by explicitly modeling the character- and word-level features of phishing attacks [5]. The Texception network [6], designed by Microsoft to classify phishing attacks, is an effective modification of a CNN, showing the best performance in supervised learning-based phishing URL classification tasks. The convolution operation aims to learn a spatial filter to extract features in the local receptive field that shares weights [7], and the long short-term memory (LSTM), a variant of an RNN, is a memory cell that stores the weights used for mapping between inputs and outputs [8].

Despite the successful development of a deep-learning-based phishing URL classifier, on the other hand, the supervised learning approach, which focuses on minimizing the loss of the classification performance relying on a large number of observations and known attacks, addresses the limitations in the detection of phishing attacks. The main difficulty, expressed as a zero-day phishing attack [9], is that phishing URLs are generated and discarded immediately after the information is stolen.

In Figure 1, we visualize the benign and phishing URL space classified by existing supervised machine-learning techniques using URL observations collected from the PhishTank [10] database. The blue and red dots represent benign and phishing URL observations, respectively, and the colored areas represent the decision boundary of the supervised classifiers. From the nature of the zero-day attack, we note that the confusion of the classifier occurs due to the class imbalance issue in which benign URLs are observed extremely less than phishing URLs.

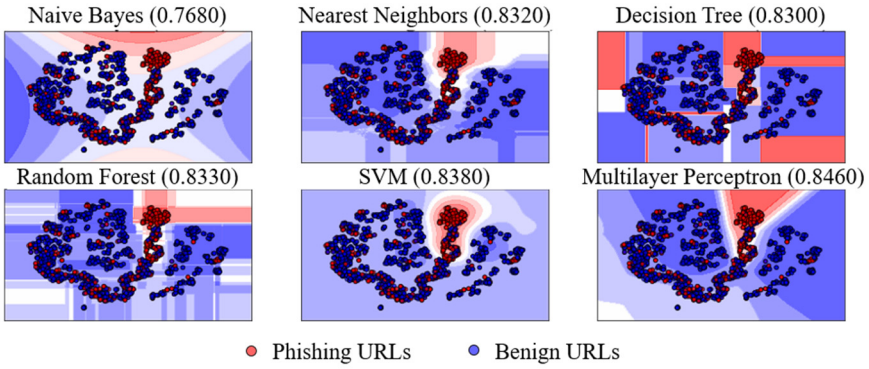


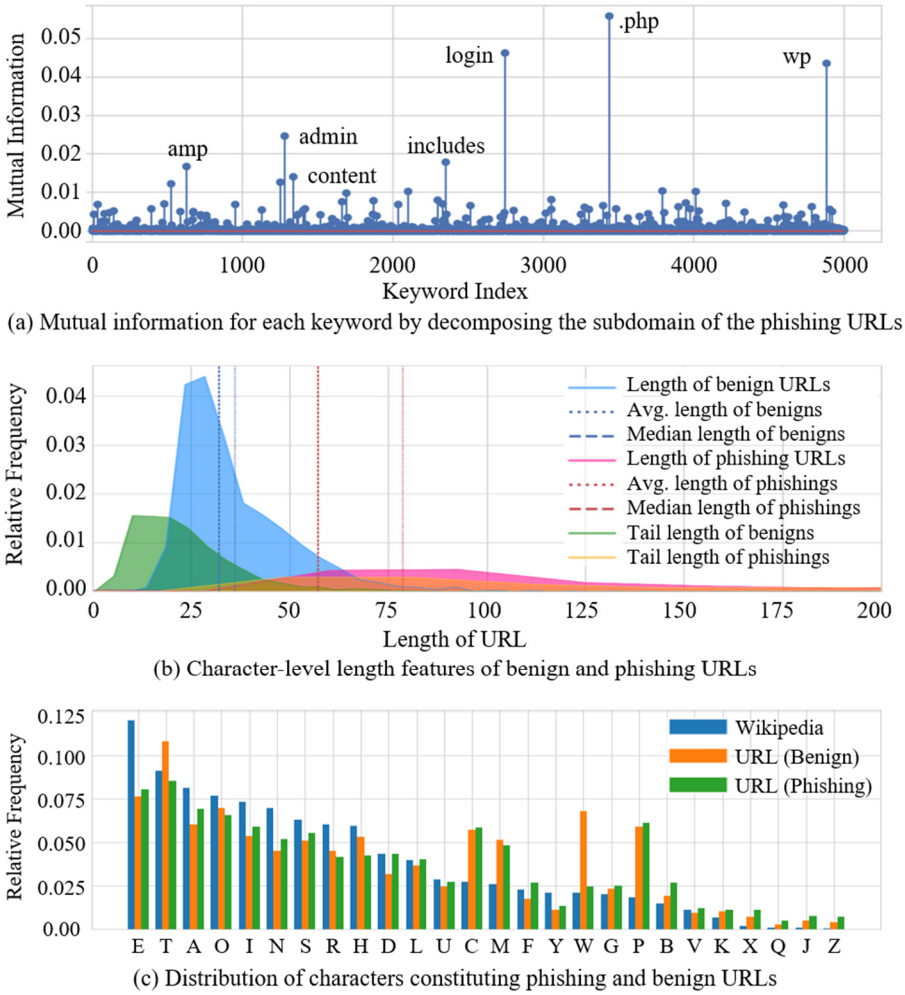
Figure 1. Class imbalance issue addressing the limitations of the supervised learning approach in the field of phishing URL modeling.

Furthermore, we visualized three major statistics in the distribution of characters that consist of URLs in Figure 2 to focus on the difficulties inherent in the field of URL modeling. In Figure 2a, we quantified the effect of specific subdomains on the characteristics of phishing URLs as mutual information. As security experts point out, it was confirmed that keywords such as wp, admin, and content from default settings in the personal server and php keyword can be used as abnormal features of phishing URLs. Figure 2b,c shows that phishing URLs are particularly longer than benign URLs and have a composition that is significantly different from the alphabetic distribution constituting natural language.

Taken together, the simultaneous consideration of the nature of a zero-day attack and character-level characteristics of phishing URLs is a promising approach that can ensure both sustainability and intelligibility in the phishing URL detection task. We noted that in order to satisfy the two requirements, an anomaly detection framework that can cope with class imbalance [11] and an optimized operation for URL modeling [12] are essential.

In this paper, we propose a combination of a convolution operation to model the character-level URL features and a deep autoencoder (AE) to consider the nature of zero-day attacks. The main innovation of this study is the introduction of an anomaly detection framework for phishing URL detection based on a convolutional autoencoder (CAE). Unlike the supervised learning approach, we have an advantage in that we constructed a URL template by learning the autoencoder with only benign URLs. We defined the abnormal score of phishing URLs by utilizing the characteristics of the autoencoder, which reduces reconstruction performance for unobserved data. Moreover, we learned an auxiliary convolutional neural network that can improve the sensitivity of the detection by using the phishing abnormal score as a feature of the phishing URL. Extensive experiments on three real-world datasets consisting of 222,541 URLs showed the highest performance among the latest deep-learning methods. In order to demonstrate the superiority of the proposed method, we performed receiver-operating characteristic (ROC) curve analysis in addition to 10-fold cross-validation and confirmed that the accuracy improved by 3.98%

compared to a comparative study [6], and the sensitivity improved by 2.83% in the hostile class-imbalanced condition.



(a) Mutual information for each keyword by decomposing the subdomain of the phishing URLs

(b) Character-level length features of benign and phishing URLs

(c) Distribution of characters constituting phishing and benign URLs

Figure 2. Three main statistics supporting the strong need for character-level modeling in the phishing URL detection task: (**a**) mutual information by keyword; (**b**) availability of the URL length feature; (**c**) character distribution that separates benign and phishing URLs.

The main contribution of this paper is that we formulated phishing detection as an anomaly detection problem with a severe class-imbalanced condition and solved it efficiently by extending the existing deep autoencoder. To the best of our knowledge, this is the first attempt where a convolutional autoencoder is incorporated to reconstruct a URL and measure the abnormal score for a phishing attack. The main findings of this research can be summarized as follows:

- The convolutional autoencoder works well for modeling the deep representation of benign URLs, resulting in the best accuracy for phishing detection.
- The abnormal score defined based on the reconstruction error of the autoencoder is suitable for the phishing detection, resulting in a significant improvement in recall.

2. Related Works

In this section, we review the relevant phishing URL detection methods based on machine-learning algorithms. The phishing URL detection research can be categorized into URL representation, modeling methods, and learning methods as summarized in Table 1.

Table 1. Related works on machine-learning-based phishing URL detection with respect to the learning approach and URL representation.

Approach	URL Representation	Method	Dataset	Author
Supervised	Text	Transition diagram	PhishStorm	Liu [2]
	Selected URL features	Rule-based detection algorithm	PhishTank	Azeez [13]
		Hierarchical classifier based on feature group	PhishTank	Mohammad [14]
		Machine-learning methods: AdaBoost, BN, Decision Table, NB, RF, etc.	UCI phishing Huddersfield URL	Osho [15]
	Text	Machine-learning methods: NB, DT, RF, SVM	UCI phishing	Chiew [16]
	Char.-/word-level embedding	URLNet: CNN concatenation	VirusTotal	Le [5]
	Selected URL features	LSTM-based text GAN	PhishTank	Anand [17]
	URL with screenshots	Optical character recognition with ResNet, homomorphic encryption	MS URL screenshots	Chou [18]
	Char.-/word-level embedding	Texception: CNN concatenation with Inception	MS anonymized URLs	Tajaddodanfar [6]
Weakly supervised	Text	Adversarial label learning	PhishTank	Arachie [19]
Anomaly detection	Char.-level encoding	Denoising autoencoder	PhishTank, hpHost	Yan [20]

As an initial attempt to model phishing attacks, malware epidemiology was proposed and implemented with diagram-based compartment models [2]. Azeez et al. proposed a simple rule-based detection algorithm utilizing the four characteristics of suspicious URLs and exploited the URL characteristics effective for classification based on the classification performance [13]. Mohammad et al. contributed to the automation of the phishing URL detection task by systematically extracting URL features and proposing a hierarchical classifier according to the extraction rule [14,21]. The URL features collected and refined for phishing classification were fully exploited for 35 machine-learning-based classifiers, including the unfamiliar methods in Osho et al., and achieved a classification performance of 0.9570 based on the random forest algorithm [15].

On the other hand, as it was revealed that the rule-based URL feature selection and modeling has a limitation in the generalization performance for unobserved URLs [22], machine-learning-based [23] phishing detection was actively studied and reached better performance. Naive Bayes (NB), decision tree (DT), random forest (RF), Bayesian network (BN), and support vector machine (SVM) were quantitatively evaluated to model phishing URLs [16], and it was emphasized that the nonlinear mapping function was effective according to the natural language characteristics in the URL. Moreover, the phishing URL database [2,21,24] that stores the observed phishing attacks provides an ideal testbed for the deep-learning-based URL classification task with a relatively closed environment. Various deep-learning methods such as CNN [5,6] and its modification [18,25,26] are proposed, as well as the LSTM-based generative adversarial network (GAN) [17] for exploiting the class imbalance issue by generating phishing URLs.

The majority of the current research in deep-learning-based phishing detection focuses mainly on optimizing the operation of the neural network. In particular, the comparative study in [6] proves the superiority of the modified 1D-convolution operation with variable filter size compared with several competitors. This motivates our decision to consider the anomaly detection-based approach proposed in this paper. The learning method is mainly categorized into four approaches: a supervised approach that learns the phishing

URL feature and its selection method directly from the label classification result, a semi- or weakly supervised approach that uses only a small number of labels or noisy labels to consider the realistic constraints [19], and an unsupervised approach that does not use label information [27] of URLs, and an autoencoder-based anomaly detection approach [20]. The fact that phishing URLs are not used to learn the benign URL model in the unsupervised approach is an advantage in class imbalance conditions and, more importantly, is an amenable solution for modeling the nature of a zero-day attack.

3. Proposed Method

In this section, we describe the combination of the convolutional autoencoder with an auxiliary classifier that learns the threshold function to detect the phishing URL based on the anomaly detection framework. Figure 3 illustrates the overall architecture of the proposed method, which consists of URL preprocessing steps, a character-level deep URL model based on an autoencoder, and phishing URL detection based on the abnormal score and URL reconstruction representing the phishing URL features.

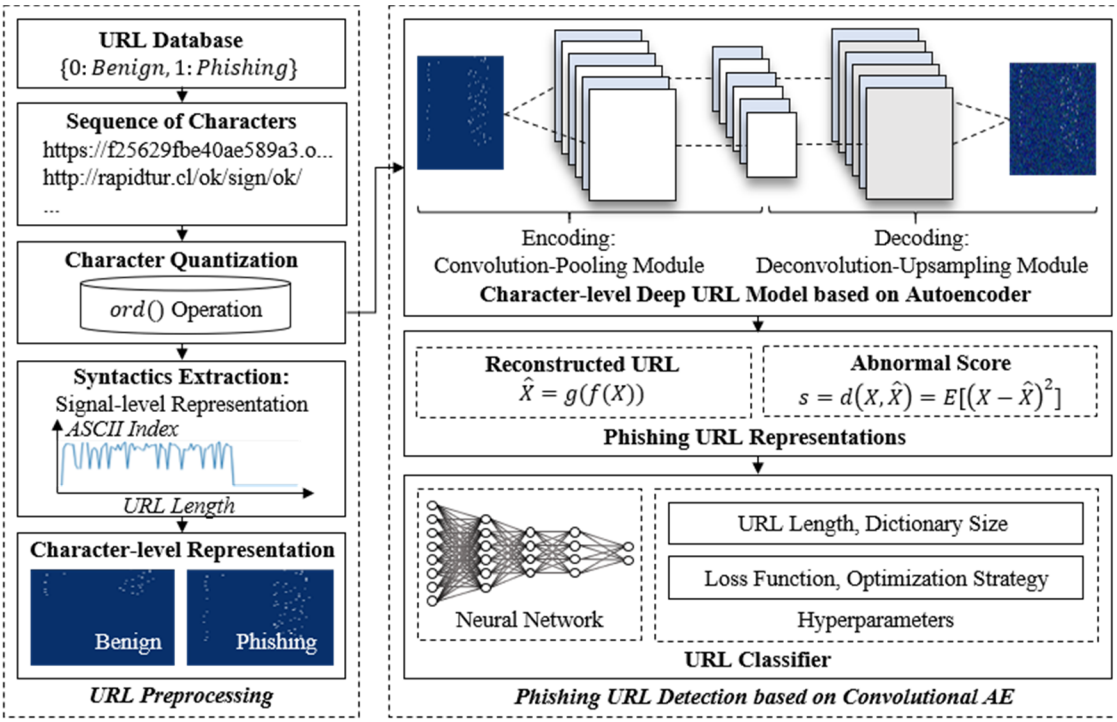


Figure 3. Two major components for phishing URL detection: a convolutional autoencoder for constructing a deep URL model and an auxiliary convolutional neural network for learning the threshold function.

3.1. Character-Level URL Model Based on a Convolutional Autoencoder

We performed two preprocessing steps to focus on the syntactics of the URL described in Section 1. The first is to allocate a unique integer to the characters that constitute the URL. We simply implemented the allocation function by extracting the ASCII code using the built-in Python function *ord*( ) [28]. The second is the one-hot encoding of each code to remove the arithmetic relationship from the sequence of integers. We encoded each character from URLs by replacing each alphabet with 1-of-m predefined integers. We defined the character dictionary as 26 alphabets, 10 numbers, and 54 to 64 special characters, including whitespace, and encoded, as shown in Figure 4. Three benchmark phishing URL datasets were preprocessed, and 100 characters were cropped in consideration of the average length of URLs in each dataset. URLs shorter than the 100-character limit were zero padded. Finally, the *i*th observed URL x_i of $X = [x_1, \ldots, x_n]$ forms a vector of size (length, size of dictionary).

␣!"#$%&'()*+,-./0123456789:;<=>?@
[]^_`abcdefghijklmnlpqrstuvwxyz{|}~...

Figure 4. A total of 90 to 100 characters to be encoded as integers, including 26 alphabets, 10 numbers, and 54 to 66 special characters, including whitespace.

The general idea of an autoencoder is to represent the data through a nonlinear encoder to a hidden layer and use the hidden units as the new feature representations, as depicted in Figure 5 [29,30]:

$$h^l = \sigma\left(W^1 x_i + b^1\right);\ \hat{x}_i = \sigma\left(W^2 h^l + b^2\right) \tag{1}$$

where $h^l \in \mathbb{R}^z$ is the URL representation of lth layer, and $\hat{x}_i \in \mathbb{R}^d$ is interpreted as a reconstruction of a normalized input URL $x_i \in \mathbb{R}^d$. The parameter set includes weight matrices $W^1 \in \mathbb{R}^{z \times d}$ and $W^2 \in \mathbb{R}^{d \times z}$ and bias vectors $b^1 \in \mathbb{R}^z$ and $b^2 \in \mathbb{R}^d$ with dimensionality z and d, and $\sigma(\cdot)$ is a nonlinear activation function. The core idea is to maximize the reconstruction error for the unobserved URL instance by learning the autoencoder using only benign URLs and to implement encoding function $f(\cdot)$ and decoding function $g(\cdot)$ with a convolutional neural network to fully exploit the character-level URL features [31].

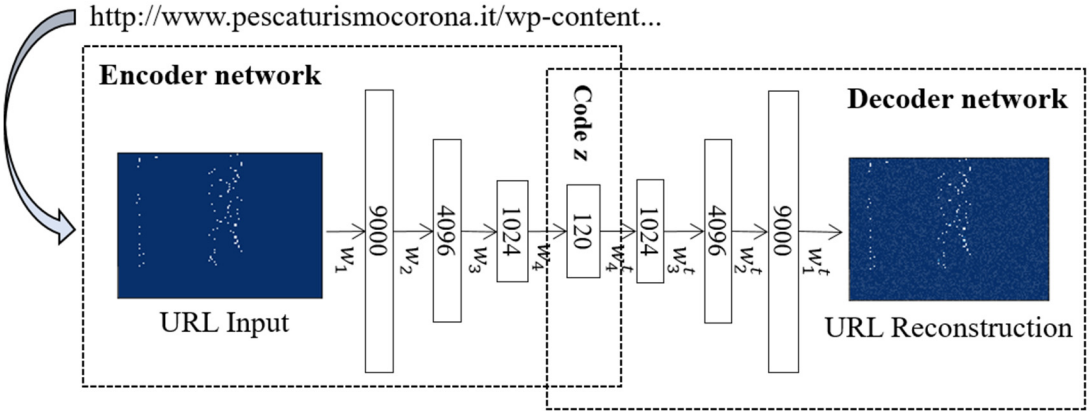


Figure 5. An illustration of autoencoder that reconstructs a URL image expressed as a vector of $(length,\ dictionary\ size)$. The URL image is encoded as a 120-dimensional vector in the hidden layer.

The major hurdle in modeling the URL with a neural network lies in extracting the spatial features from the limited URL samples [32,33]. We construct the convolutional layer and the deconvolutional layer for learning the benign URL features from the convolutional autoencoder. It is well known that the convolution operation has advantages represented by data-driven filter learning focused on extracting spatial features in the field of pattern recognition [34,35].

The convolution operation $\phi_C(\cdot)$ and the max-pooling operation $\phi_P(\cdot)$ in CNNs, which have been successfully applied for extracting the character-level features, are suitable to model the sequence of characters in URLs and extract the features using local connectivity between characters [36]. The convolution operation is known to reduce the translational variance between features [37,38] and preserves the spatial relationship between URL characters by learning filters to extract the hidden correlations. Given the $(k \times k)$-sized filter W of the lth convolutional layer, the stacked convolutional operation is applied with the input URL x^l_{mn} in the row m and the column n:

$$\phi^l_C(x_i) = \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} w^l_{ab} x^{l-1}_{(m+a)(n+b)} \tag{2}$$

Because the dimension of the output vector that has been distorted and copied by the convolution operation $\phi_C(\cdot)$ is increased by the number of convolution filters, the

summary statistics from nearby node activation are extracted from max-pooling $\phi_P(\cdot)$. Pooling refers to a dimensionality reduction process from the $(k \times k)$ region in order to impose the capacity bottleneck and facilitate faster computation [39]:

$$\phi_P^l(x_i) = \max x_{mn\times\tau}^{l-1} \tag{3}$$

The convolutional autoencoder has been extensively utilized in the field of anomaly detection and novelty detection. It compresses and reconstructs character-level URL features through an encoding function $f_\theta(\cdot)$ consisting of the convolution/pooling operation and decoding function $g_\theta(\cdot)$ performing an inverse operation. We define a convolutional autoencoder to construct a deep URL model with the reconstructed URL $\hat{x}_i$ for the compressed URL code h:

$$\hat{x}_i = g_\theta(f_\theta(x_i)) = \phi_C^{-1}\left(\phi_P^{-1}(\phi_P(\phi_C(x_i)))\right) \tag{4}$$

The distance function between the inputted benign URL x_i and the reconstructed URL $\hat{x}_i$ can be implemented with the Euclidean distance, and the loss function of the convolutional autoencoder is defined as the error between input x_i and reconstructed $\hat{x}_i$:

$$L_{MSE}(\theta; X,\ \hat{X}) = \sum_i (x_i - \hat{x}_i)^2 \tag{5}$$

The objective of autoencoder learning is to find the encoding/decoding parameter θ that minimizes the loss function L_{MSE}, and we trained the network using the backpropagation algorithm based on the stochastic gradient descent method according to the basic neural network training method:

$$\begin{aligned} \theta^* &= argmin_\theta \textstyle\sum_{x_i \in X_{benigns}} L_{MSE}(\theta; X,\ \hat{X}) \\ &= argmin_\theta \textstyle\sum_{x_i \in X_{benigns}} \sum_i (x_i - \hat{x}_i)^2 \end{aligned} \tag{6}$$

3.2. Phishing URL Classification Based on Reconstruction Errors

Because only benign URLs are used for the learning of the convolutional autoencoder, exploiting the parameter θ^* that optimizes the reconstruction of the benign URL means that it is difficult to reconstruct phishing URLs with different character distributions and length characteristics. According to the traditional autoencoder-based anomaly detection framework, we defined an abnormal score S_τ with threshold τ and distance function $d(,)$ based on the reconstruction error:

$$s_\tau(x_i) = d(x_i, \hat{x}_i) = \|x_i - \hat{x}_i\|_2^2 \tag{7}$$

The distance function $d(\cdot,\cdot)$ can be implemented as a Manhattan distance or a cosine distance, but we defined it as the most intuitive Euclidean distance by referring to the loss function of a convolutional autoencoder.

The abnormal score S_τ defined for the reconstruction URL $\hat{x}_i$ can be used as a classifier by applying a thresholding rule for itself. However, we constructed an additional phishing URL classifier for the reconstruction URL $\hat{x}_i$, as it was known that the thresholding rule is limited in generalization performance for unobserved instances or phishing URLs similar to the benign URL distribution. Because the input of the auxiliary classifier that finally performs the phishing classification task is the reconstructed URL image, we implemented the classifier $\phi(\cdot)$ using a convolutional neural network. Intuitively, the convolutional neural network learns a thresholding function that classifies labels from reconstructed URL images with weight matrices $W \in \mathbb{R}^2$:

$$\hat{y}_i = softmax(\phi(\hat{x}_i)) = softmax(\phi_P(\phi_C(\sigma(W\hat{x}_i + b)))) \tag{8}$$

Finally, the objective of the auxiliary classifier that learns the thresholding function is to find the parameter θ that minimizes the loss function L_{CE} implemented with cross-entropy between predicted and actual label:

$$L_{CE}(\theta; Y, \hat{Y}) = -\sum_i y_i \log(\hat{y}_i) \quad (9)$$

4. Experimental Results

In this section, we present how the convolutional autoencoder with character-level embedded URLs predicts the phishing attack and evaluate the performance with 10-fold cross-validation in terms of accuracy and recall [40], which is followed by quantitative comparison with the latest deep-learning models.

4.1. Dataset and Implementation

Here we validate the proposed convolutional autoencoder and auxiliary classifier that utilizes the reconstruction error as an abnormal score with the benchmark URL database. For extensive evaluation, three real-world URL datasets consisting of 222,541 benign and phishing URLs were collected and are summarized in Table 2. The ISCX-URL-2016 dataset aims at the four-way classification task consisting of benign, phishing, malware, and spam URLs and has a 3:1 class imbalance as a characteristic of malicious URL modeling. Web-accessible Phishstorm and Phishtank datasets provide known phishing attack cases. Unlike the Phishstorm dataset where class sampling was performed, Phishtank does not provide a benign URL. We collected benign URLs from the Open Directory Project and collected 95,541 and 60,000 URLs.

Table 2. Source and description of the three benchmark phishing URL datasets.

Source	URL Label	Instances	e.g., (Accessed Date: 19 October 2020)
ISCX-URL-2016 [21]	Benign	35,000	http://metro.co.uk/2015/05
	Phishing	9000	http://standardprincipal.pt/
	Malware	11,000	http://9779.info/%E5%88%
	Spam	12,000	http://adverse*s.co.uk/scr/cl
PhishStorm [24]	Benign	47,682	en.wikipedia.org/wiki/Walkingdead
	Phishing	47,859	nobell.it/70ffb52d079109dc
PhishTank [10]	DMOZ Open Directory Project (Benign)	45,000	http://geneba**.org/ftp/
	OpenDNS (Phishing)	15,000	http://droopbxoxx.com/@@@

The architecture of the convolutional autoencoder can be modified variously according to the number of stacked convolution and pooling layers, as well as the number of convolutional filters, the kernel size, and the number of the nodes in layers. Given that typical deep-learning models require an optimization process, it is essential to adjust and optimize the hyperparameters carefully. A total 3,677,115 of deep-learning hyperparameters of the proposed method were determined through an empirical trial and error of the iterative optimization process. The number of convolutional filters indicating the number of local reception fields for learning spatial features between URL characters, the size of a reception field, stride as a parameter of overlapping regions, the type of an activation function of the layer, and the number of layer-by-layer parameters are specified in Table 3.

Table 3. Summary of the hyperparameters of the convolutional autoencoder.

Operation	No. of Convolution Filters	Kernel Size	Stride	Activation Function	No. of Parameters
Reshape2D	-	-	-	-	0
Convolution 2D	256	2×2	1	tanh	1280
MaxPooling 2D	-	2×2	2	-	0
Convolution 2D	512	2×2	1	tanh	524,800
MaxPooling 2D	-	2×2	2	-	0
BatchNormalization	-	-	-	-	2048
Convolution 2D	1024	1×1	1	tanh	525,312
Upsampling 2D	-	-	-	-	0
Deconvolution 2D	512	2×2	1	tanh	2,097,664
Upsampling 2D	-	-	-	-	0
Deconvolution 2D	512	2×2	1	tanh	524,544
BatchNormalization	-	-	-	-	1024
Convolution 2D	1	2×2	1	tanh	257
Reshape 2D	-	-	-	-	0

4.2. Phishing Detection Performance

Table 4 compares the accuracy and recall for the latest deep models, including the standard deep-learning networks (CNN, LSTM) and their major modifications, which achieve state-of-the-art results. CNN and CNN-LSTM used as the base network achieved a 0.9424 accuracy and 0.9015 recall in the ISCX-URL-2016 dataset. We assumed URLNet, which achieved the best performance in URL classification by using a CNN in parallel, and Microsoft's Texception network, which improved the inception operation in the CNN for the URL field, as major comparative studies. The Texception network achieved accuracies of 0.9765, 0.9710, and 0.9319 for each dataset, but URLNet composed of a vanilla CNN achieved a similar level of performance for CNN and CNN-LSTM. Surprisingly, the triplet network structure, which has recently attracted much attention in the field of signal processing and image classification, and its modification, the Monte Carlo search-based triplet network, achieved robust performance. The triplet network is the latest implementation of metric learning that explicitly learns the distributions of a dataset, and we note that it is relatively suitable for modeling character-level URL images.

The proposed method outperforms the latest deep-learning model. As argued, it was effective to model both class imbalance and character-level features in the URLs, and we achieved the highest accuracy and recall in all three benchmark datasets. On the other hand, the thresholding attempt based on the anomaly score calculated from URL reconstruction without an auxiliary classifier showed performance degradation.

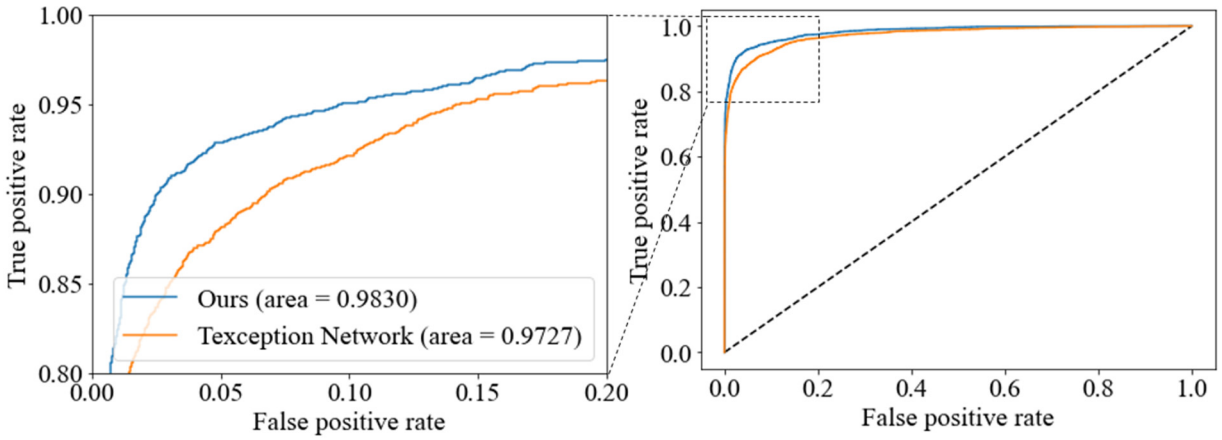
In Figure 6, receiver-operating characteristic (ROC) analysis was conducted to show the improvement of the recall with the comparative study. The x- and y-axes represent the false positive rate and the true positive rate for the output of the phishing URL classifier, respectively, and our approach to learning the thresholding function produced an area-under-the-curve (AUC) improvement of 1.06%.

We compared the proposed method and comparative study in terms of accuracy and recall under severe class imbalance conditions in Figure 7. The class imbalance ratio was adjusted along the x-axis while removing the phishing URL from the training dataset from Phishtank based on the assumption of a zero-day phishing attack situation. The imbalance ratio is the number of phishing URLs compared to the benign URLs scaled in the [0.0,1.0] range. For example, at an imbalance ratio of 1.0, it is assumed that there is no phishing URL instance in the training dataset. For a fair comparison, we applied a class weight algorithm that was proportional to the number of data when training two networks.

Initially, both the proposed method and the Texception network showed accuracies of 0.9642 and 0.9635, but the accuracy degraded linearly as the number of phishing URL instances decreased. Because the proposed methods include the thresholding mechanism based on the abnormal score, a classification accuracy of 0.8883 was achieved even in the severe class-imbalanced condition.

Table 4. Comparison of 10-fold cross-validation in terms of accuracy and recall.

Benchmark Dataset	**ISCX-URL-2016**		**PhishStorm**		**PhishTank**	
Metrics	**Acc.**	**Recall**	**Acc.**	**Recall**	**Acc.**	**Recall**
Base Network						
Character-CNN [31]	0.9363 ± 0.0060	0.8909 ± 0.0212	0.9016 ± 0.0042	0.8565 ± 0.0352	0.8852 ± 0.0120	0.8034 ± 0.0294
LSTM	0.9175 ± 0.0166	0.8803 ± 0.0181	0.8777 ± 0.0219	0.8440 ± 0.0277	0.8544 ± 0.0242	0.7865 ± 0.0383
CNN-LSTM [8]	0.9424 ± 0.0057	0.9015 ± 0.0147	0.9229 ± 0.0142	0.8785 ± 0.0183	0.9070 ± 0.0084	0.8374 ± 0.0244
Comparative Studies						
URLNet [5]	0.9450 ± 0.0043	0.9390 ± 0.0110	0.9395 ± 0.0050	0.8864 ± 0.0192	0.9226 ± 0.0123	0.8785 ± 0.0212
Texception [6]	0.9765 ± 0.0049	0.9462 ± 0.0097	0.9710 ± 0.0031	0.9227 ± 0.0187	0.9319 ± 0.0108	0.9075 ± 0.0114
Triplet Network [41]	0.9505 ± 0.0122	0.9064 ± 0.0227	0.9473 ± 0.0085	0.8902 ± 0.0249	0.9081 ± 0.0221	0.8469 ± 0.0274
Monte Carlo Search based Triplet Net. [37]	0.9673 ± 0.0133	0.9227 ± 0.0282	0.9664 ± 0.0071	0.9065 ± 0.0265	0.9237 ± 0.0237	0.8665 ± 0.0285
Convolutional Autoencoder-based Phishing Detection (Proposed)						
Thresholding using Anomaly Score	0.9734 ± 0.0035	0.9338 ± 0.0085	0.9532 ± 0.0041	0.9091 ± 0.0166	0.9120 ± 0.0071	0.8655 ± 0.0221
Threshold Learning with Auxiliary CNN	0.9780 ± 0.0027	0.9590 ± 0.0074	0.9732 ± 0.0018	0.9338 ± 0.0131	0.9690 ± 0.0084	0.9132 ± 0.0185



(**a**) Comparison in the rage of true positive rates [0.8, 1.0] (**b**) Comparison over the full range

Figure 6. Receiver-operating characteristic (ROC) analysis to show the improvement of recall with the comparative study. (**a**) Comparison of ROC curves in the range of true positive rates [0.8, 1.0]; (**b**) Comparison in [0.0, 1.0] range of ROC curve of Texception Network.

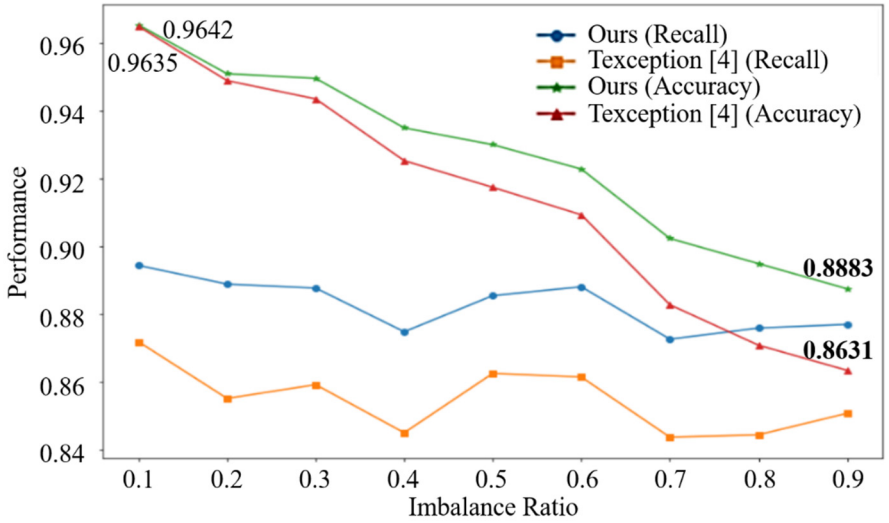


Figure 7. Robustness of accuracy and recall by class imbalance ratio.

4.3. Performance Evaluation by Component: URL Reconstruction and Effect of the Auxiliary Classifier

We conducted confusion matrix analysis to verify the effect of the auxiliary classifier in the Phishtank dataset in Table 5. In parentheses, the result of thresholding-based classification using the anomaly score is described, excluding the auxiliary classifier, which utilizes the URL reconstruction from the convolutional autoencoder. Referring to the statistics of misclassified cases that deviate from the main diagonal matrix, we confirmed an improvement in recall and accuracy in both benign and phishing URLs.

Table 5. Confusion matrix analysis to verify the effectiveness of the auxiliary classifier.

Confusion Matrix		**Predicted (w/o Auxiliary Classifier)**		
		Benign	**Phishing**	**Recall-**
Actual	Benign	8854 (8377)	187 (781)	0.9793 (0.9147)
	Phishing	281 (824)	2678 (2018)	0.9050 (0.7101)
	Precision	0.9692 (0.9104)	0.9347 (0.7209)	Accuracy: 0.9610 (0.8663)

The convolutional autoencoder, which is the core idea of the proposed method, is optimized for reconstructing benign URLs. We compared the input and reconstructed images for normal and phishing URLs in Figure 8. The white dots in the URL image represent characters, and the sequence of characters in the URL is recorded along the y-axis. In the benign URL, there is little visual difference between the input and the reconstructed URL, whereas the phishing URL has a blurring effect. There was no significant difference in terms of the structural similarity index (SSIM), which measures the difference in distribution instead of the pixel difference in the image; however, in terms of the root mean square error (RMSE), which actually measures the Euclidean distance between pixels, we confirmed the increased reconstruction error for the phishing URL.

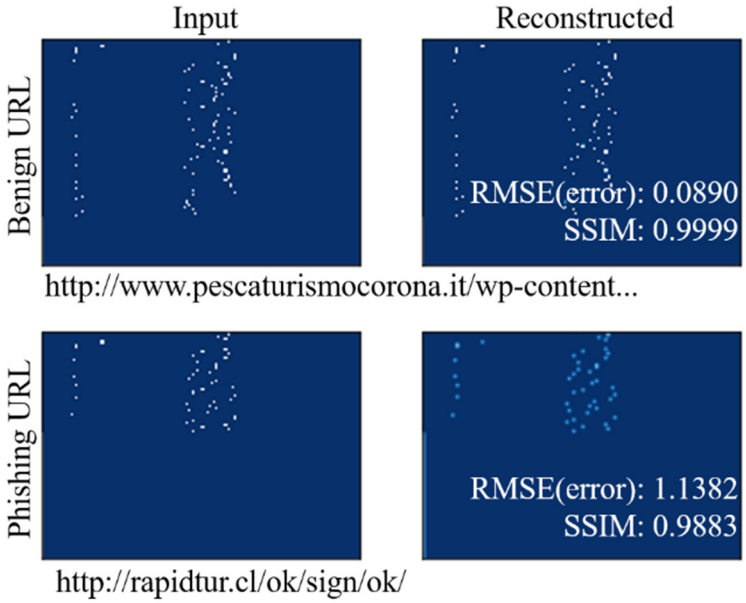
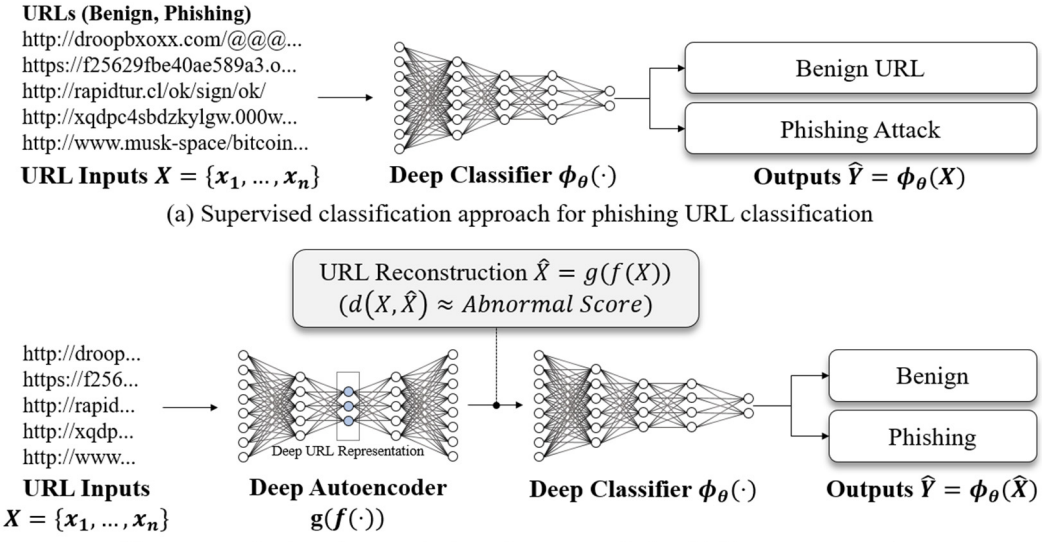


Figure 8. Comparison of the URL and its reconstructions with benign and phishing URLs.

4.4. Discussions

We compared the supervised approach and the proposed autoencoder-based anomaly detection approach in Figure 9. The deep-learning-based URL classifier, which achieved the highest performance so far, as described in Figure 9a, focuses on minimizing the classification errors to learn the parameter θ defined as a set of the weights of a neural network. On the other hand, in the proposed method described in Figure 9b, there is an explicit step of modeling a benign URL before classification. Considering the autoencoder learns the encoding/decoding operation to reconstruct the output, the reconstruction performance degrades for inputs with different distributions (mainly phishing URLs) after learning with only the benign URLs.



(a) Supervised classification approach for phishing URL classification

(b) Autoencoder-based anomaly detection approach for phishing URL Detection

Figure 9. Anomaly detection approach that can construct a template for benign URLs and measure the abnormal score for a phishing attack based on the abnormal score measured by the autoencoder.

We visualized the decision boundary mentioned in Section 1 to understand the pros and cons of the proposed method. The deep URL representation generated from the hidden

layer of the convolutional autoencoder was mapped into a two-dimensional space [42] using the t-SNE algorithm, as depicted in Figure 10, and the main misclassified case was extracted from the area of the top-right where the classifier is confused. Correctly classified cases at the top and bottom of both sides were also extracted and are listed in Table 6. We confirmed that the correctly classified cases fully support the research hypothesis that the syntactics from the sequence of characters in a URL should be exploited. As a case in which the anomaly score increased significantly, there was a phishing URL composed of a sequence of random characters, and the case that fits the benign URL distribution output a low anomaly score, as expected.

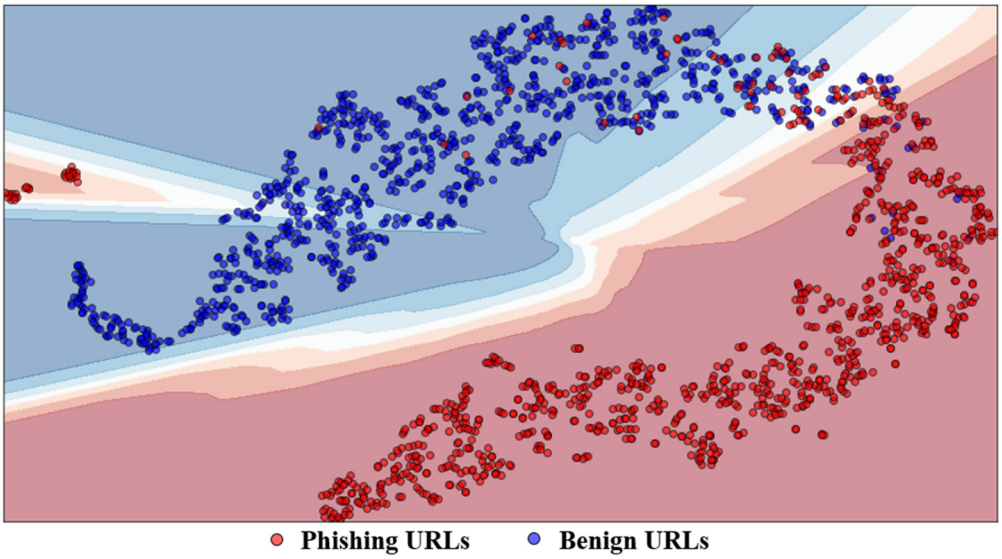


Figure 10. Decision boundary of URL representation extracted from the hidden layer of the proposed convolutional autoencoder.

Table 6. Summary of the correctly and misclassified cases.

Classification Result	Label	URL (Accessed Date: 19 October 2020)	Anomaly Score s_τ
Correctly classified	Phishing	https://1drv.ms/xs/s!AhtvzT3KrwqMZzLMKnTc8clHnRA? wd-FormId=%7BA0F7982D%2D71A4%2D4DE0%2DB4C4%2DC16A	0.9874
	Benign	http://market.security***.net	0.0031
Misclassified	Benign	http://archives.seattletimes.nwsou***.com/cgi-bin/texis.cgi/web/vortex/display?slug=will&date=199903	0.8815
	Phishing	http://tesla-present.site/ethereum/	0.0584

However, the benign URL is misclassified by phishing when a long and noisy sequence of characters is observed. Several readable phishing URLs are misclassified as benign. This misclassified case suggests that additional URL features to be utilized remain, although the proposed method achieves the best performance among the deep models. Considering the fact that CNN was used in parallel in the comparative study and modeled not only character-level but also word-level URL features, the limitation can be handled by extending the proposed method with an additional convolution operation for the full utilization of URL characteristics.

5. Concluding Remarks

In this study, we proposed a character-level convolutional autoencoder based on the anomaly detection framework to overcome the two difficulties of phishing URL detection. The main innovation of this study is the introduction of deep anomaly detection to the field of phishing URL detection and achieving the best performance compared to

classification-based deep-learning methods by implementing a neural network structure and an operation optimized for URL modeling. The combination of the encoding/decoding structure to facilitate disentanglement between classes and convolution operation optimized for character-level URL characteristics was utilized to define an anomaly score based on the reconstruction error.

The limitation of the proposed methodology is that it was optimized for character-level features among the various features constituting URLs. We discussed that the confusion of the character-level features is the main cause of the performance degradation of the proposed method. Considering the structure of the web address consisting of domains and subdomains, additional performance improvements can be expected by utilizing the word-level features, including the typos and the keywords listed in the blacklist.

In a future study, we can consider the additional exploitation of URL features to improve the detection performance. At the same time, an additional convolution operation that utilizes both the character- and word-level URL features is required to fully exploit URL features. We also suggest exploring a plausible solution to zero-day attacks, which can be expressed as an out-of-distribution issue. Considering that the features that are not exposed in the dataset can be modeled from the external knowledge of domain experts, it would be promising to introduce a symbolic AI approach that leverages the detection rules based on the domain knowledge into the field of phishing URL detection.

Author Contributions: Conceptualization, S.-B.C.; formal analysis, S.-J.B.; funding acquisition, S.-B.C.; investigation, S.-J.B.; methodology, S.-J.B. and S.-B.C.; supervision, S.-B.C.; visualization, S.-J.B.; writing—review and editing, S.-B.C. Both authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01361, Artificial Intelligence Graduate School Program (Yonsei University)).

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

References

1. Lastdrager, E.E. Achieving a consensual definition of phishing based on a systematic review of the literature. *Crime Sci.* **2014**, *3*, 9. [CrossRef]
2. Liu, W.; Zhong, S. Web malware spread modelling and optimal control strategies. *Sci. Rep.* **2017**, *7*, 42308. [CrossRef]
3. Yang, C.; Harkreader, R.; Gu, G. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1280–1293. [CrossRef]
4. Fazil, M.; Abulaish, M. A hybrid approach for detecting automated spammers in twitter. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2707–2719. [CrossRef]
5. 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.
6. 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.
7. Bu, S.-J.; Cho, S.-B. A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. *Inf. Sci.* **2020**, *512*, 123–136. [CrossRef]
8. Bu, S.-J.; Cho, S.-B. Time Series Forecasting with Multi-Headed Attention-Based Deep Learning for Residential Energy Consumption. *Energies* **2020**, *13*, 4722. [CrossRef]
9. Souri, A.; Hosseini, R. A state-of-the-art survey of malware detection approaches using data mining techniques. *Hum. Cent. Comput. Inf. Sci.* **2018**, *8*, 3. [CrossRef]
10. 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.
11. Andresini, G.; Appice, A.; Malerba, D. Autoencoder-based deep metric learning for network intrusion detection. *Inf. Sci.* **2021**, *569*, 706–727. [CrossRef]
12. Wei, W.; Ke, Q.; Nowak, J.; Korytkowski, M.; Scherer, R.; Woźniak, M. Accurate and fast URL phishing detector: A convolutional neural network approach. *Comput. Netw.* **2020**, *178*, 107275. [CrossRef]
13. Azeez, N.A.; Salaudeen, B.B.; Misra, S.; Damaševičius, R.; Maskeliūnas, R. Identifying phishing attacks in communication networks using URL consistency features. *Int. J. Electron. Secur. Digit. Forensics* **2020**, *12*, 200–213. [CrossRef]

14. Mohammad, R.M.; Thabtah, F.; McCluskey, L. An assessment of features related to phishing websites using an automated technique. In Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions, London, UK, 10–12 December 2012; pp. 492–497.
15. Osho, O.; Oluyomi, A.; Misra, S.; Ahuja, R.; Damasevicius, R.; Maskeliunas, R. Comparative Evaluation of Techniques for Detection of Phishing URLs. In Proceedings of the International Conference on Applied Informatics, Madrid, Spain, 7–9 November 2019; pp. 385–394.
16. Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.; Tiong, W.K. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Inf. Sci.* **2019**, *484*, 153–166. [CrossRef]
17. Anand, A.; Gorde, K.; Moniz, J.R.A.; Park, N.; Chakraborty, T.; Chu, B.-T. Phishing URL detection with oversampling based on text generative adversarial networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 1168–1177.
18. Chou, E.J.; Gururajan, A.; Laine, K.; Goel, N.K.; Bertiger, A.; Stokes, J.W. Privacy-Preserving Phishing Web Page Classification Via Fully Homomorphic Encryption. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2792–2796.
19. Arachie, C.; Huang, B. Adversarial label learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3183–3190.
20. Yan, H.; Zhang, X.; Xie, J.; Hu, C. Detecting Malicious URLs Using a Deep Learning Approach Based on Stacked Denoising Autoencoder. In Proceedings of the Chinese Conference on Trusted Computing and Information Security, Wuhan, China, 18 October 2018; pp. 372–388.
21. Mamun, M.S.I.; Rathore, M.A.; Lashkari, A.H.; Stakhanova, N.; Ghorbani, A.A. Detecting malicious urls using lexical analysis. In Proceedings of the International Conference on Network and System Security, Taipei, Taiwan, 28–30 September 2016; pp. 467–482.
22. Iuga, C.; Nurse, J.R.; Erola, A. Baiting the hook: Factors impacting susceptibility to phishing attacks. *Hum. Cent. Comput. Inf. Sci.* **2016**, *6*, 8. [CrossRef]
23. Om, K.; Boukoros, S.; Nugaliyadde, A.; McGill, T.; Dixon, M.; Koutsakis, P.; Wong, K.W. Modelling email traffic workloads with RNN and LSTM models. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*, 1–16. [CrossRef]
24. Marchal, S.; François, J.; State, R.; Engel, T. PhishStorm: Detecting phishing with streaming analytics. *IEEE Trans. Netw. Serv. Manag.* **2014**, *11*, 458–471. [CrossRef]
25. Burnap, P.; French, R.; Turner, F.; Jones, K. Malware classification using self organising feature maps and machine activity data. *Comput. Secur.* **2018**, *73*, 399–410. [CrossRef]
26. Vasan, D.; Alazab, M.; Wassan, S.; Safaei, B.; Zheng, Q. Image-based malware classification using ensemble of CNN architectures (IMCEC). *Comput. Secur.* **2020**, *92*, 101748. [CrossRef]
27. Qin, Z.-Q.; Ma, X.-K.; Wang, Y.-J. ADSAD: An unsupervised attention-based discrete sequence anomaly detection framework for network security analysis. *Comput. Secur.* **2020**, *99*, 102070. [CrossRef]
28. Yuan, B.; Wang, J.; Liu, D.; Guo, W.; Wu, P.; Bao, X. Byte-level malware classification based on markov images and deep learning. *Comput. Secur.* **2020**, *92*, 101740. [CrossRef]
29. Xayasouk, T.; Lee, H.; Lee, G. Air Pollution Prediction Using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) Models. *Sustainability* **2020**, *12*, 2570. [CrossRef]
30. Sureda Riera, T.; Bermejo Higuera, J.-R.; Bermejo Higuera, J.; Martínez Herraiz, J.-J.; Sicilia Montalvo, J.-A. Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review. *Sustainability* **2020**, *12*, 4945. [CrossRef]
31. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 649–657.
32. Yang, P.; Zhao, G.; Zeng, P. Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access* **2019**, *7*, 15196–15209. [CrossRef]
33. 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.
34. Jang, U.; Suh, K.H.; Lee, E.C. Low-quality banknote serial number recognition based on deep neural network. *J. Inf. Process. Syst.* **2020**, *16*, 224–237.
35. Wen, J. Gait recognition based on GF-CNN and metric learning. *J. Inf. Process. Syst.* **2020**, *16*, 1105–1112.
36. Bu, S.-J.; Cho, S.-B. A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Oviedo, Spain, 20–22 June 2018; pp. 561–572.
37. Bu, S.-J.; Park, N.; Nam, G.-H.; Seo, J.-Y.; Cho, S.-B. A Monte Carlo Search-Based Triplet Sampling Method for Learning Disentangled Representation of Impulsive Noise on Steering Gear. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3057–3061.
38. Ni, S.; Qian, Q.; Zhang, R. Malware identification using visualization images and deep learning. *Comput. Secur.* **2018**, *77*, 871–885. [CrossRef]
39. Er, M.J.; Zhang, Y.; Wang, N.; Pratama, M. Attention pooling-based convolutional neural network for sentence modelling. *Inf. Sci.* **2016**, *373*, 388–403. [CrossRef]

40. Pei, X.; Yu, L.; Tian, S. AMalNet: A deep learning framework based on graph convolutional networks for malware detection. *Comput. Secur.* **2020**, *93*, 101792. [CrossRef]
41. Novoselov, S.; Shchemelinin, V.; Shulipa, A.; Kozlov, A.; Kremnev, I. Triplet Loss Based Cosine Similarity Metric Learning for Text-independent Speaker Recognition. In Proceedings of the Interspeech, Hyderabad, India, 2–6 September 2018; pp. 2242–2246.
42. Carrasco, R.S.M.; Sicilia, M.-A. Unsupervised intrusion detection through skip-gram models of network behavior. *Comput. Secur.* **2018**, *78*, 187–197. [CrossRef]