HCNN-LSTM: Hybrid Convolutional Neural Network with Long Short-Term Memory Integrated for Legitimate Web Prediction

Candra Zonyfar[1], Jung-Been Lee[1,2] and Jeong-Dong Kim[1,2,3,*]

[1]*Department of Computer Science and Electronics Engineering, Sun Moon University, Korea*
[2]*Department of Computer Science and Engineering, Sun Moon University, Asan, 31460 South Korea*
[3]*Genom-Based BioIT Convergence Institute, Sun Moon University, Korea*
E-mail: candrazonyfar@sunmoon.ac.kr; jungbini@sunmoon.ac.kr; kjdvhu@gmail.com
[*]*Corresponding Author*

Received 04 August 2023; Accepted 14 October 2023;
Publication 19 December 2023

Abstract

Phishing techniques are the most frequently used threat by attackers to deceive Internet users and obtain sensitive victim information, such as login credentials and credit card numbers. So, it is important for users to know the legitimate website to avoid the traps of fake websites. However, it is difficult for lay users to distinguish legitimate websites, considering that phishing techniques are always developing from time to time. Therefore, a legitimate website detection system is an easy way for users to avoid phishing websites. To address this problem, we present a hybrid deep learning model by combining a convolution neural network and long short-term memory (HCNN-LSTM). A one-dimensional CNN with a LSTM network shared estimation of all sublayers, then implements the proposed model in the benchmark dataset for phishing prediction, which consists of 11430 URLs with 87

Journal of Web Engineering, Vol. 22_5, 757–782.
doi: 10.13052/jwe1540-9589.2251

attributes extracted of which 56 parameters are selected from URL structure and syntax. The HCNN-LSTM model was successful in binary classification with accuracy, precision, recall, and F1-score of 95.19%, 95.00%, 95.00%, 95.00%, successively outperforming the CNN and LSTM. Thus, the results show that our proposed model is a competitive new model for the legitimate web prediction tasks.

Keywords: Phishing detection, cyber threat, CNN-LSTM, deep learning, machine learning.

1 Introduction

The mitigation of internet security threats posed by phishing crimes continues to present an ongoing challenge for the research community [1–5]. Phishing, a form of cybercrime, poses considerable risks to unsuspecting users who encounter cloned web pages mimicking authentic websites. These deceptive replicas can lead to the inadvertent transmission of sensitive user data to malicious servers, thereby compromising the security and privacy of individuals [6, 7].

Phishing websites are serious cybercrime threat that needs to be addressed because there have been many reports of losses due to this crime. Based on the data [8, 9], a staggering 96% of phishing attacks are delivered through email. These malicious emails often employ the tactic of impersonating reputable companies or organizations, deceiving unsuspecting victims into unwittingly revealing their login credentials, credit card details, and other sensitive information. The FBI's reports indicate losses of over USD$1.8 billion due to business email compromises and an additional US$29.1 million from ransomware attacks data [10]. Phishing attacks leverage both technical and non-technical methods to deceive users into divulging sensitive information, and the effectiveness of the attack solely relies on the victim's ability to differentiate between a legitimate and a phishing website. Phishing attackers clone legitimate websites, create fake domain names, and send convincing phishing links to users, aiming to obtain sensitive data for financial fraud or resale after users interact with the fake interface. As technology develops, attackers have made various adaptations to various anti-phishing methods. As with implementing SSL protection, currently, 78% or more of phishing websites implement SSL protection that is used exclusively by legitimate websites [11]. This shows that hackers are increasingly using modern and sophisticated methods.

Unfortunately, it is not easy for users to detect phishing crimes, which increase from time to time; moreover according to data in [12], a new fake phishing website is launched every 20 seconds. So an automatic phishing prediction system will greatly assist users in distinguishing between legitimate websites and phishing websites. To tackle the issue, this study proposes a legitimate website detection model, named HCNN-LSTM, that is a hybrid of a deep learning (DL) model by combining CNN and LSTM networks.

The remaining parts of this work are divided into several sections. Section 2 provides a basis and related works. Section 3 addresses the phishing website detection approach. In Section 4, the experiment is presented. Section 5 provides the conclusion and future works.

2 Related Work

2.1 Phishing-attacks Context

The way phishing works is not affected by the strength of the password algorithm or the sophistication of the firewall algorithm. Because, basically, the crime of phishing is only trapping, meaning that if someone can realize the difference between a phishing website and a legitimate website, they will avoid losses due to phishing cybercrimes. Attackers take at least two approaches to carry out their actions against phishing targets. First is the technical aspect, relating to computer engineering capabilities such as creating phishing websites by cloning legitimate websites, setting URL domains, etc. Second, non-technical aspects, the attacker takes an emotional approach in this section on how the target or user can be influenced to do what the attacker expects. Clicking on URL links (sometimes even in the form of images, buttons, or QR codes so that users do not get suspicious), then performing activities such as filling out forms, logging in, forgetting passwords, and so on. Because when the user/target clicks the submit button, they send their data to the malware server. A detailed explanation is shown in Figure 1.

Figure 2 shows the cycle of phishing website attacks [5]. First, the attacker designs the pages of the phishing website by cloning the legitimate website and creating a very similar interface between the phishing website and the legitimate website. Attackers also create fake domain names that, at first glance, look like real websites. This domain name/URL usually has incorrect characters and spelling. For naive users, it can be difficult to distinguish between a fake URL name and a legitimate web URL name. User navigation tends to focus on the interface, where all the website content has

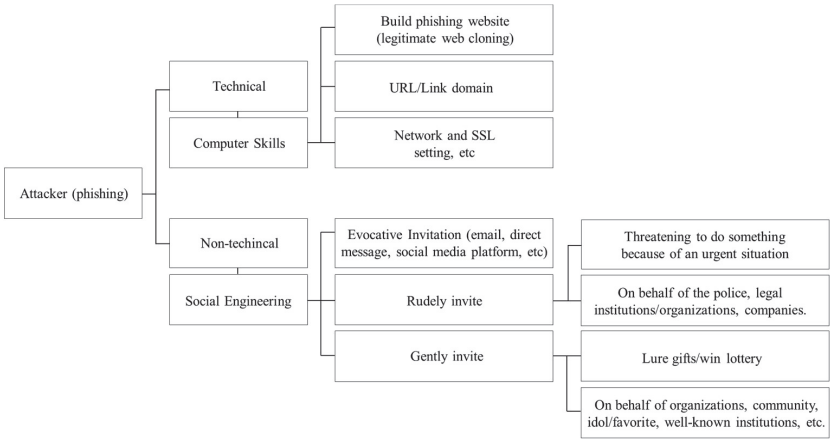

Figure 1 Common approaches for phishing website attackers.

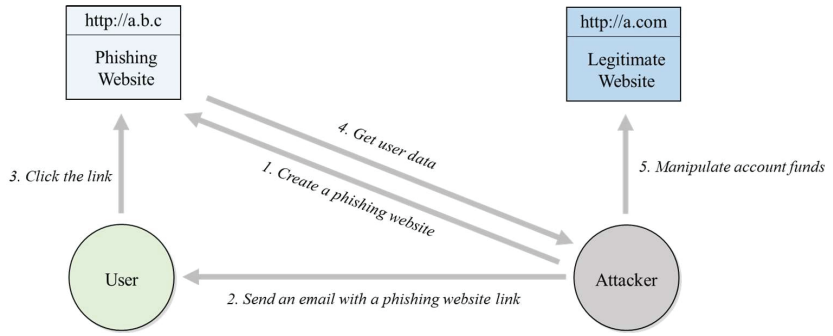

Figure 2 Phishing life cycle.

been designed to resemble the original website, such as login page, logo, layout, font types, etc. The attacker then sends the generated phishing website link to the user, usually sent via email. However, as technology advances, many attackers take advantage of platforms such as social media or instant messaging. When a user has been on a phishing website, all the data and information written on the phishing website form will point to the malware server [13]. At that time, the attacker obtains the credential data such as usernames, passwords, personal information, bank accounts, etc. The purpose of the attacker doing phishing is to get user data and information, which then the attacker can use for various purposes. They make it possible to use login credentials and passwords to make financial transactions, for example, or they can also collect data to offer then sell, which can happen and makes sense.

2.2 Existing Studies on Phishing Detection

We have entered the era of machine learning (ML), where computers can learn to solve problems independently by looking for patterns in large data. A specific problem can be addressed by looking for hidden patterns in the data set and then outlining possible solutions. A simple explanation is that the ML model is presented as a data set, which then analyses the data and discovers hidden tendencies to answer predictions. Every correct or incorrect prediction that is made is stored and used as a guide for future responses [14, 15]. As a result, ML gets better with practice. Table 1 shows research related to phishing detection with various ML approaches.

On the other hand, DL systems can learn from their own experiences, but they often operate best with large datasets. One of the most recent applications of DL is speech processing, image recognition and natural language processing [24]. DL can use long input sample sequences to improve its capabilities. Identifying at least two unique aspects of this particular DL strategy is possible. DL can learn hierarchical and complex feature representations with ease. Successive DL can also use the previous input sequence data as a starting point for new calculations [25].

In recent years, scientists have performed a wide number of investigations on the proposed approaches for identifying criminal threats on the internet [4, 19, 26–28]. The DL approach has been utilized frequently in various text-related problems, including text categorization and machine translation [22, 29], treating the URL string as a series of text for detecting phishing sites.

Table 1 Various ML research for web phishing detection

Authors	Models	Datasets
Gupta et al., 2021 [16]	Feature important, ML (RF, KNN, SVM, LR)	Phishing dataset
Apoorva et al., 2022 [17]	XGBoost, LightGBM, AdaBoost, Gradient boosting	Openphish, PhishTank
Rao et al., 2021 [18]	RF	Self-collected
Paniagua et al., 2022 [19]	LGBM	PILWD-134K
Liew et al., 2019 [20]	RF	PhishTank
Hannousse et al., 2021 [3]	Feature selection, ML (DT, RF, LR, NB, SVM)	Self-collected
Hussain Et al., 2023 [21]	CNN-Fusion	DS-1, Ebbu2017, PhishStorm, ISCX-URL2016, DS-5
Zheng et al., 2022 [22]	HDP-CNN	PhishTank, Alexa web
Oram et al., 2021 [23]	LGBM	Phishing dataset for ML

DL techniques are used to understand the representation of URL functionality at the dictionary and semantic levels.

Phishing website detection research has been conducted using various DL and DL approaches. Studies using ML have been carried out, such as those conducted by Hannousse and Yahiouche [3] who did two steps in this phishing website detection research. First, based on the unavailability of a phishing website benchmark dataset, which evaluation is considered fair and because phishing methods are constantly evolving, so need to be adapted quickly too. Hannousse and Yahiouche [3] proposed a general scheme for building reproducible datasets in the future. Second, Hannousse and Salima Yahiouche [3] experimented with several decision tree algorithms, random forest, logistic regression, SVM, and naive Bayes. Meanwhile, the research of Chiew et al. [30] used WEKA and compared their proposed method, hybrid ensemble feature selection (HEFS), with several other ML methods, including SVM, naive Bayes, and C4.5.

Research on the detection of phishing websites using another ML approach was carried out by Adebowale et al. [31] who proposed using multimodal, using feature images, frames and text. As a classifier, Adebowale et al. [31] experimented with ML models such as KNN, SVM, and ANFIS as their proposed method. They used two different datasets, namely The University of California Irvine, Rami et al., [32] and the University of Huddersfield, Rami et al. [33]. This research obtained image data from the sources PhishTank [34] and Anti-Phishing Working Group (APWG) [11]. Sahingoz et al. [35] researched phishing website predictions by creating a dataset from PhishTank for phishing website data and legitimate website data sourced from the Yandex Search API [36]. A total of 73.575 data were collected, which contained 36.400 legitimate website data and 37.175 phishing website data. This study uses WEKA, where at least the dataset has experimented with 7 ML algorithms, decision tree, AdaBoost, kStar, KNN, random forest, SMO, and naive Bayes, with the best results obtained with the random forest algorithm.

Paniagua et al. [19] offer a dataset consisting of 134.000 sample data which they named the Phishing Index Login Websites Dataset (PILWD). This research consists of stages: collecting datasets, then determining the phishing index login website dataset (here considering several things: phishing samples, login pages, legitimate samples, login pages). The next stage is collecting data and extracting handcraft features. They rely on URL, HTML, Hybrid, and tech at this extraction stage. The last stage is classification. Paniagua et al. [19] explained that the hybrid feature covered aspects

(copyright in the HTML, domain in HTML, domain-copyright, subdomain-copyright, path-copyright, title-domain-copyright, title-domain, domain in body, subdomain in the title). As for tech features, this refers to aspects of the number of technologies detected and specific technologies. For the tech feature, they used a wappalyzer report. In the prediction section, they utilized the XGBoost ML approach, LightGBM being the best performing model, AdaBoost, random forest, kNN, SVM, logistic regression, and naive Bayes. This latest study concludes that it excels in several respects compared to previous phishing website detection studies, such as the dataset year aspect, the total of the valid sample data is 66.964, which, compared to others, is less than 60.000. Also, the number of sample data for phishing websites is 66.964, whereas other phishing detection studies are less than 40.668.

On the other hand, a phishing website prediction study has also been carried out using a DL approach. Alshehri et al. [37] used 10.234 URLs sourced from PhishStorm phishing dataset in their research on predicting phishing websites. Where 7.234 total websites are legitimate, and 3.004 are phishing website data. Then they did training with a ratio of 80:20 using the CNN model. The CNN architecture consists of five convi1D layers, flattened and concatenated, then three dense layers and one dropout layer, and ends with a sigmoid activation layer. Zeng et al. [22] proposed a method called the highway deep pyramid CNN (HDP-CNN). They reported four modules start from the embedding stage, highway network, region embedding and finally, the deep pyramid. The embedding region consists of convolutional layers with sizes [2–5]. As experiments, they collected as much as 420.000 data from sources PhishTank and Alexa. The data was then divided into 70:15:15 for training, validation, and testing data.

Recent research on web phishing detection proposes a DL multi-scale semantic deep fusion models approach [38]. The authors propose a method of detecting phishing websites with multiple semantic functional blocks of webpages in their experiment. They consider URL, title, body text, and invisible text input, then do text vectorization and word embedding. Then the multiple convolutional layers and the max pooling layer are concatenated before the fully connected layer. In this research, various scenarios of kernel number [3, 5, 7, 9, 11] and dropout rate [0,1,0.3,0.5,0.7,0.9] were carried out. Another recent study experimented with ML and DL methods, namely decision tree, ANN, random forest, gradient boosting, fully connected feedforward deep neural networks, LSTM and CNN.

Meanwhile another new study works [39] with data on 60.252 websites, of which 32.972 are legitimate websites and a total of 27.280 phishing

websites. This study focuses on features (URL, textual content, hyperlinks, login form information). Meanwhile, they utilize ML models for training and testing, including eXtreme gradient boosting (XGBoost), random forest, logistic regression, naive Bayes, and an ensemble of random forest Adaboost classifier, where the best performance is obtained from the XGBoost classifier model by considering combining all the features.

2.3 Convolution Neural Network (CNN)

CNN is a DL model that has achieved great success in various tasks, such as pattern recognition, classification, object detection, and segmentation [21, 40–42]. Although CNN was originally developed for image processing, because of its ability to extract features hierarchically, the CNN architecture has been adapted and successfully applied to text data for natural language processing (NLP) assignments [29]. CNN can retrieve local features from text by using convolution operations as it is done on images so that it allows CNN to find important patterns in sentences or text locally.

The CNN primarily consists of two main components, namely the convolutional layer and the pooling layer [22]. The convolutional layer is responsible for applying convolutional operations to the input data, allowing the network to detect various features and patterns within the data. On the other hand, the pooling layer serves to reduce the spatial dimensions of the data, thereby reducing the computational complexity and extracting the most relevant information from the previously learned features. Together, these two layers play a crucial role in enabling CNNs to effectively learn hierarchical representations from the input data and make them well-suited for tasks such as image recognition and feature extraction [21, 22].

2.4 Long Short-term Memory (LSTM) Neural Network

LSTM is a DL model that has been widely used in classification work related to security [41, 43–46]. LSTM was developed to address the limitations of back-propagation in traditional RNN. It is one of the most reliable algorithms for handling sequential data because of its adaptability to long-term or short-term dependencies [44]. Within this model, three gates are present, namely the input gate, output gate, and forget gate [45]. These gates are governed by the following equations:

$$f_t = \sigma_{\mathrm{g}}(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma_{\mathrm{g}}(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$

$$o_t = \sigma_{\mathrm{g}}(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

$$h_t = 0_t \odot \sigma_h(c_t) \tag{6}$$

where f_t, i_t, and o_t represent the forget, input, and output gates correspondingly. W_f, W_i, W_c, and W_o signify the weight matrices associated with these gates. U_f, U_i, U_c, and U_o pertain to the recurrent connections to these gates. The functions σ_{g}, σ_c, σ_h and $\odot$ denote sigmoid, hyperbolic tangent, hyperbolic tangent, and element-wise multiplication, respectively.

3 Proposed Method

This section presents a proposed hybrid CNN-LSTM model for detecting legitimate websites. We have developed a DL model designed for classification tasks, with the primary objective of effectively distinguishing between legitimate URLs and phishing URLs. The key components of the proposed model comprise CNN and LSTM blocks. Therefore, we define the task as a classification challenge. The dataset consists of URL strings paired with labels, where each label represents either a legitimate web or a phishing web. We encode the numerical value 1 to indicate web legitimate and the value 0 to indicate web phishing. In general, we can express research questions of this binary classification tasks using the notation depicted in the following equation [22]:

$$D = \{(a, b) | a = a_j,\ b = b_j,\ j \in Z\} \tag{7}$$

where D is the representation of the dataset, a denotes the URL string, $b \in \{0, 1\}$ denotes the associated tag (0 for a legitimate web URL string and 1 for a phishing web string), and Z represents the size of the dataset. Here, a neural network is utilized to learn the feature representation of URLs and predict phishing websites, denoted as $b'_j = t(b_j)$, where t represents a function transformation based on the model. This formulation problem is based on previous studies in the domain of phishing detection, specifically binary classification tasks performed by Yan et al., [22] in the context of phishing detection.

The HCNN-LSTM model combines both CNN and LSTM to process hierarchical data structures, where sequences are embedded within

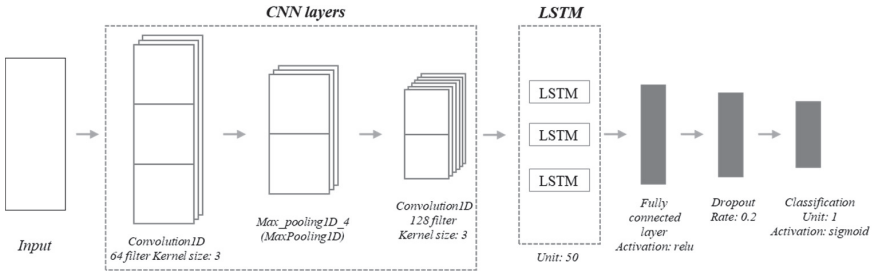

Figure 3 Architecture of the HCNN-LSTM.

higher-level contexts. This model is particularly useful when dealing with structured data that exhibits various levels of hierarchy. Regarding the use of URLs in this model, the URL string itself is not directly utilized as input. Instead, attributes extracted from the URL are employed as features for the model. These attributes are in the form of component features obtained from the URL. More details about features along with examples of their features are explained in Section 4.1.

Figure 3 shows the contextual model of the HCNN-LSTM architecture. The HCNN-LSTM follows a sequential layer arrangement, starting with a convolutional 1D layer designed to extract local features. Each convolutional layer uses filter sizes of 64 and 128 respectively, with a kernel size of 3, which facilitates the identification of important patterns. A 1D MaxPooling layer with a pool size of 2 is inserted after each convolutional layer to reduce representation.

Subsequently, long short-term memory (LSTM) layers were incorporated, enabling the model to capture temporal dependencies within the data. Three LSTM layers were sequentially stacked, each containing 50 units. The initial two LSTM layers were configured to return sequences, facilitating the propagation of information across time steps. The final LSTM layer, acting as a sequence-to-vector converter, summarized the temporal information.

The model concludes with a dense layer, employing a sigmoid activation function. This layer offers a nonlinear transformation to the extracted features. The output layer consists of a single neuron, suitable for binary classification tasks. Overall, this intricate architecture combines convolutional and recurrent components, enabling the model to capture both spatial and temporal patterns inherent in the data. The compiled model is ready for training with the optimizer, positioning us to iteratively fine-tune the weights and biases.

The success of CNN algorithm development can be attributed to its ability to learn new patterns from data features. CNN's most significant feature is the convolutional layer, which implements a convolution kernel to combine inputs. It acts as a filter and is then turned on by a non-linear activation function, as follows:

$$h_t = \sigma(w_{i,h} \cdot x_t + w_{h,h} \cdot h_{t-1} + b) \tag{8}$$

$$a_{i,j} = f\left(\sum_{m=1}^{M}\sum_{n=1}^{N} w_{m,n} \cdot x_{i+m.j+n} + b\right) \tag{9}$$

where $a_{i,j}$ represents the corresponding activation, $w_{m,n}$ represents the $m \times n$ weight vector of the convolution kernel, $x_{i+m.j+n}$ represents the activation of the higher neurons connected to neuron (i, j), b represents the bias parameter.

The results from the CNN process are forwarded to the four LSTM layers to extract temporal features from sequence data more effectively. To control the activity of each memory cell, input is sent to several gates, including input gates, forget gates, and output gates. The output layer of LSTM-CNN model is composed of a fully connected (FC) layer and a SoftMax classifier. The addition of the FC layer towards the conclusion of the algorithm has significant benefits. Each node of the FC layer is connected to the nodes of the top layer to integrate the retrieved attributes from the upper layer.

Behind the FC layer is the sigmoid classifier, which transforms the output of the higher layer into a probability vector whose value indicates the probability that the given sample belongs to a certain class. The equation for the expression is as follows:

$$S_j = \frac{e^{aj}}{\sum_{k=1}^{N} e^{ak}} \tag{10}$$

where N is the number of classes and a, j is the output vector jth value. With conv1D layer as pre-processing for the LSTM layer. Then layer max-pooling1D with a pool size = '2' and conv1D filter layer 128 with a kernel size = '3'. The LSTM layer, a FC layer with ReLU activation and dropout layer = '0.2'.

4 Experiments

This section aims to experiment the performance of our proposed model which is applied in a case study in classifying legitimacy websites and

phishing websites. The experiments are performed on an Ubuntu 18.0.4.64bit with NVIDIA GeForce RTX 1080 Ti×4 GPU and RAM 62.7GB. Models are developed in python 3.6.0 with the help of TensorFlow 2.1.5 packages, Pandas 1.1.3, and NumPy 1.19.2.

4.1 Analysis and Pre-processing Dataset

We employ the dataset provided by Hannousse et al. [10], which contains 11.430 URLs with features which are divided into three feature sections, namely URL-based features (as shown in Table 2), content-based features, and external-based features. Overall, this dataset has a total of 87 features. In URL-based features, the feature analyses the URL text in detail from a structural and statistical point of view. Table 2 describes the features, descriptions and data types of the features contained in the dataset, where the structural-based features represent the existence, position, and nature of the basic URL elements, such as the existence of ports, the use of the 'https' protocol, and the position of the top-level domain (TLD). While feature-based statistics represent the distribution of basic URL elements, certain words, or characters in URL text. Such as the number of points, subdomains, and word length. Therefore, during the pre-processing stage, after performing the encoding process, we proceed with the splitting of the dataset into training and test sets.

Prior to data pre-processing and splitting, we conducted various exploratory data analyses to gain insights into the dataset's structure and characteristics. These analyses involved examining the distribution of variables, identifying missing values, discovering patterns and relationships between variables, and identifying potential outliers and anomalies. Figure 4 presents an exploratory data analysis that visualizes the relationships between numerical features, where green indicates a positive correlation and red indicates a negative correlation. The colour intensity in the plot reflects the magnitude of the correlation. For instance, the '*long_word_path'* feature exhibits a strong positive correlation with the '*long_words_raw'* feature, and the *'length_url'* feature demonstrates high correlation with several other features, including the *'length_words_raw'* feature.

During the data coding process, we perform manipulations on the data set to improve our analytical procedures. Specifically, we apply one-hot coding to convert 'status' category data into binary, where 'legitimate' labels are assigned a value of 1 and other labels are assigned a value of 0.

Table 2 Dataset features

Feature	Description	Type
url	URL	Object
length_url	URL length	Int
length_hostname	Hostname length	Int
ip	IP address	0/1
nb_dots, nb_hyphens, nb_at, nb_qm, nb_and, nb_or, nb_eq, nb_underscore, nb_tilde, nb_percent, nb_slash, nb_star, nb_colon, nb_comma, nb_semicolumn, nb_dollar, nb_space	Special character (‘.’, ‘-’, ‘@’, ‘?’, ‘&’, ‘|’, ‘=’, ‘_’, ‘∼’, ‘%’, ‘/’, ‘*’, ‘:’, ‘,’, ‘;’, ‘$’, ‘%20’)	Int
nb_www, nb_com, nb_dslash, http_in_path	Common terms (‘www’, ‘.com’, ‘http’, ‘//’)	Int
https_token	https	0/1
ratio_digits_url, ratio_digits_host	Ratio of digits in full URLs and hostnames	Float
punycode	punycode	0/1
port	Port numbers	0/1
tld_in_path, tld_in_subdomain	TLD position	0/1
abnormal_subdomain	Abnormal subdomains	0/1
nb_subdomains	#subdomains	Int
prefix_suffix	Prefix Suffix	0/1
random_domain	Random domains	0/1
shortening_service	Shortening service	0/1
path_extension	Path extension	0/1
nb_redirection, nb_external_redirection	Redirection	Int
length_words_raw, char_repeat, shortest_words_raw, shortest_word_host, shortest_word_path, longest_words_raw, longest_word_host, longest_word_path	NLP and word-raw features	Int
avg_words_raw, avg_word_host, avg_word_path	Average length of words	Float
phish_hints	Phish hints	Int
domain_in_brand, brand_in_subdomain, brand_in_path	Brand domains	0/1
suspecious_tld	Suspicious TLD	0/1
statistical_report	Statistical report	0/1

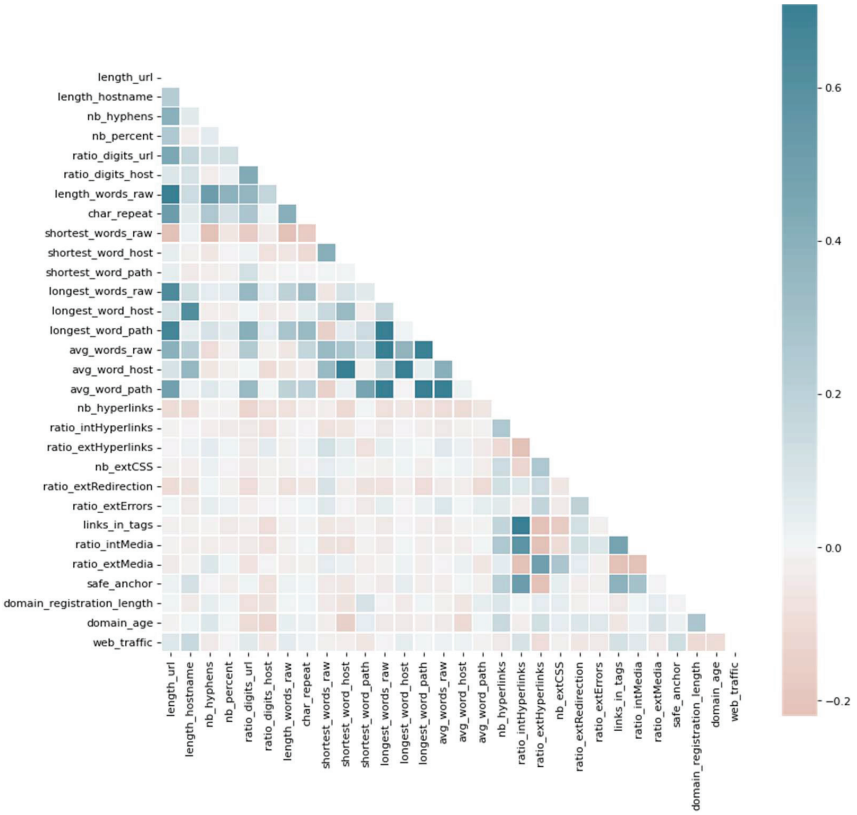

Figure 4 Correlation relationships between variables.

The splitting process between training and test sets was accomplished utilizing a stratification approach, which guarantees that the distribution of target classes in both the training and validation sets maintains the original balance. This was achieved with a size ratio of 60:40, where 40% of the data was allocated to the test set, and a total of 42 random seeds were employed to ensure reproducibility. To prepare the data for model training we extend the x train and x validation array dimensions by adding an extra dimension, which represents channel information. This step is important for compatibility with proposed models that expect channel dimensions.

In this experiment, the batch size for the LSTM, CNN, and HCNN-LSTM models was set as follows: (512, 128, 64, 32, 16, 8, 4). Among them, the best batch size for the LSTM and HCNN-LSTM models was found to be 512, while the CNN model achieved the best performance with a batch size of 64.

Table 3 Optimizable parameter of the CNN model

Layer (Type)	Output Shape	# of Parameter
conv1d_114 (Conv1D)	(None, 85, 64)	256
max_pooling1d_92(Max_Pooling1D)	(None, 42, 64)	0
conv1d_115 (Conv1D)	(None, 40, 64)	12352
max_pooling1d_94 (MaxPooling1D)	(None, 9, 64)	0
conv1d_117 (Conv1D)	(None, 7, 32)	6167
max_pooling1d_95 (MaxPoolin g1D)	(None, 3, 32)	0
activation_66 (Activation)	(None, 3, 32)	0
max_pooling1d_96 (MaxPoolin g1D)	(None, 1, 32)	0
dense_49 (Dense)	(None, 1, 1)	33
Total params: 31,169		
Trainable params: 31,169		
Non-trainable params: 0		

The LSTM architecture employed in this experiment consisted of five layers, comprising three hidden layers and one output layer. Each LSTM layer used 50 units, and a dropout rate of 0.2 was applied to each layer. For the CNN model, there were five layers with three hidden layers. Each layer utilized three kernels, and the number of filters was set to 64. The activation function 'relu' was employed. The total number of parameters for this one-dimensional CNN model was calculated to be 31,169.

To validate the developed models, we performed three model experiments and conducted a comparative analysis among them. The models included CNN, LSTM, and HCNN-LSTM, with each model's specific parameters presented in Tables 3 and 4, respectively. The first experiment involved using the CNN model. The first layer of the CNN network obtains the vector generated by the feature embedding. It has four convolution layers consisting of 64 and 32 filters, with kernel size =3. In each convolution layer, there is a max pooling with pooling number 2, which receives and processes the data before entering it into the following DL layer. Then the model classifier consists of one continuous layer, fully connected; the output layer with the 'relu' activation function. The architecture of experiment CNN model is shown in Table 3.

The second experiment utilized the LSTM model. The model is trained with fully supervision, and the gradient is sent back to the LSTM layer. Out of nine layers, four are LSTM layer number unit 50, four are dropout layers = 0.2, and the last is a fully connected layer. The rmsprop optimizer

Table 4 Optimizable parameter of the LSTM model

Layer (Type)	Output Shape	# of Parameter
lstm_36 (LSTM)	(None, 87, 50)	10400
dropout_31 (Dropout)	(None, 87, 50)	0
lstm_37 (LSTM)	(None, 87, 50)	20200
dropout_32 (Dropout)	(None, 87, 50)	0
lstm_38 (LSTM)	(None, 87, 50)	20200
dropout_33 (Dropout)	(None, 87, 50)	0
lstm_39 (LSTM)	(None, 50)	20200
dropout_34 (Dropout)	(None, 50)	0
dense_49 (Dense)	(None, 1)	51
Total params: 71,051		
Trainable params: 71,051		
Non-trainable params: 0		

Table 5 Comparison performance testing the CNN, LSTM, and HCNN-LSTM models

Classifier	Accuracy	Loss	Precision	Recall	Support	F1-scored
CNN	0.9094	0.0696	0.91	0.91	4572	0.91
LSTM	0.9486	0.0413	0.95	0.95	4572	0.95
HCNN-LSTM	**0.9519**	0.0403	0.95	0.95	4572	0.95

function is used in this LSTM experiment. The architecture of experiment LSTM model is shown in Table 4.

Finally we combined both CNN and LSTM models. This Conv1d model is compiled with the optimizer function 'RMSprop', the same as the LSTM model and the CNN-LSTM model. the HCNN-LSTM model is a combination of both the LSTM and CNN models, which were trained with a total of 300 epochs.

4.2 Results

In this paper, one public dataset was used to evaluate the generalization ability and the accuracy of the proposed method to classify of website as phishing or legitimate. The results are CNN, LSTM, and HCNN-LSTM models as shown in Table 5, with 92.96%, 95.39%, and 95.43% for each accuracy respectively with a loss value of 0.0632 for CNN, and the loss value remaining two models 0.0384. Based on the three models in the experiment, this can be seen visually in Figures 5–7.

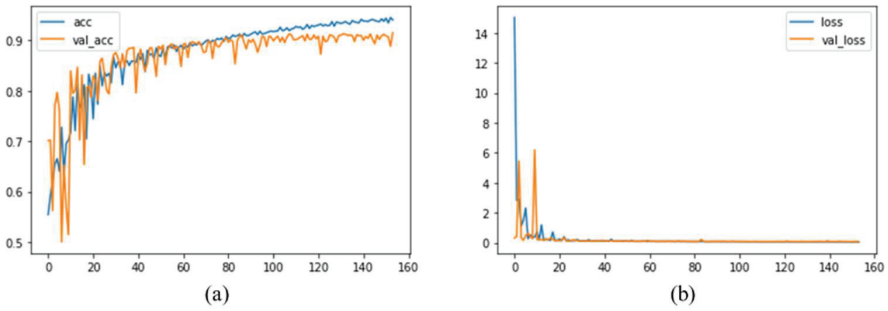

Figure 5 Results for phishing detection using the CNN model.

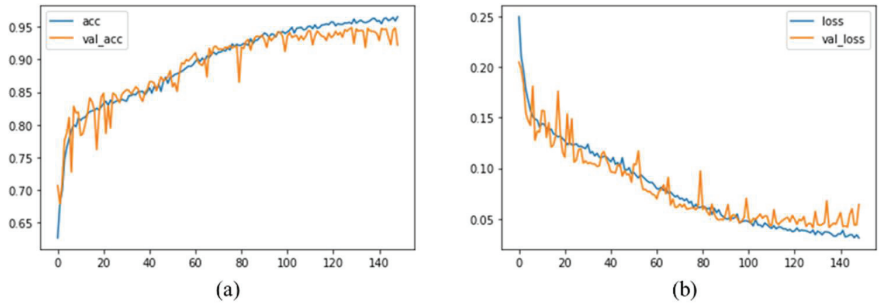

Figure 6 Results for phishing detection using the LSTM model.

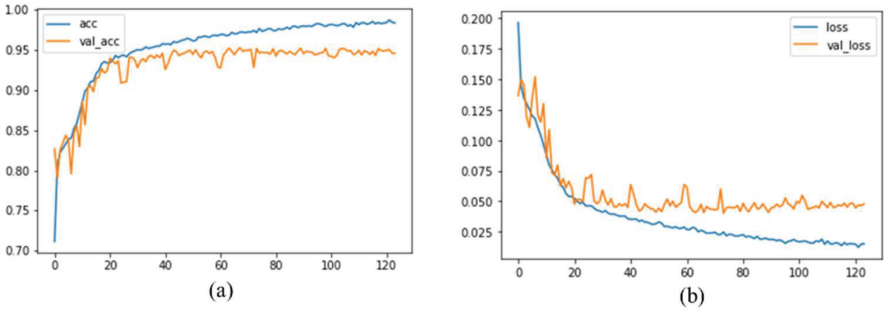

Figure 7 Results for phishing detection using the HCNN-LSTM.

Table 5 presents the findings of a comparison of the HCNN-LSTM model with those of the other two approaches. With 95.19% accuracy and a loss value of 0.0403, the combination of a CNN one dimensional embedded with LSTM produced the highest results.

The performance of DL architecture in this study is evaluated using a confusion matrix, where the phishing website class is declared a positive class, and the legitimate website class is declared a negative class. All scores are expressed in accuracy, precision, recall, and F-score. As for accuracy, according to [2, 19, 47], it is a correctly classified total sample taken as the primary metric for evaluation purposes due to its common use in phishing detection work, it can be calculated as shown in the Equations (11)–(14).

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} * 100 \tag{11}$$

$$\text{Recall} = \frac{TP}{(TP + FN)} * 100 \tag{12}$$

$$\text{Precision} = \frac{TP}{(TP + FP)} * 100 \tag{13}$$

$$\text{F1score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}. \tag{14}$$

The confusion matrix model is shown in Figure 8, where only a few data are classified incorrectly by the model. Figure 10 shows that the positive class reached the highest classification level (95.19%) for the HCNN-LSTM model, followed by the LSTM model in Figure 9, and the lowest presentation result was the CNN model in Figure 8.

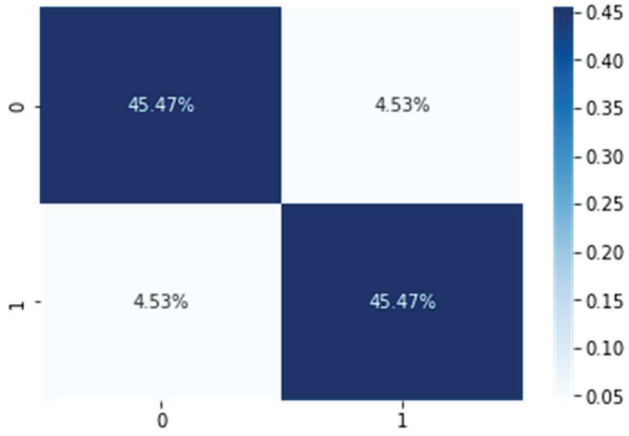

Figure 8 Confusion matrix obtained from the binary prediction level for the CNN model.

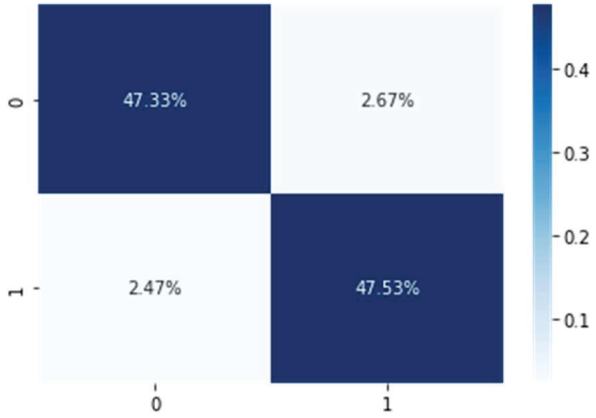

Figure 9 Confusion matrix obtained from the binary prediction level for the LSTM model.

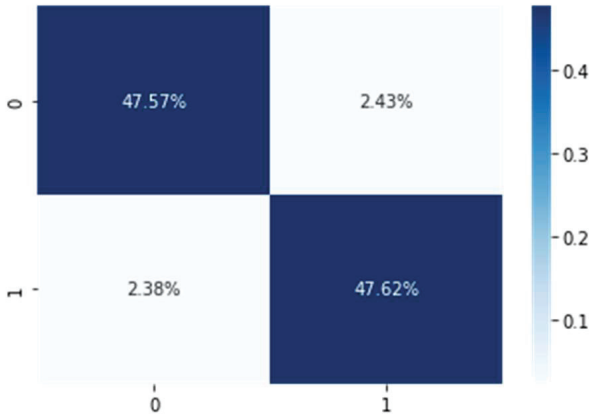

Figure 10 Confusion matrix obtained from the binary prediction level for the HCNN-LSTM model.

5 Conclusion

Among all cyber risks, web phishing crime is one of the most urgent problems and requires the attention of researchers. To prevent users from becoming victims of cyberattacks and to ensure the security of their personal and financial information online, users must ensure that their activities on the internet are carried out on a legitimate web. Unfortunately, lay users have difficulty distinguishing legitimate web and fake web. So that the legitimacy web detection system is easy for users to use, and it is very important to help improve user security. We propose HCNN-LSTM, a deep learning model for

detecting the legitimate web. The experiments were carried out to demonstrate the performance, and the accuracy obtained its high of 95.19%. CNN achieved 90.94% accuracy, although LSTM achieved 0.33% less than the proposed approach. This indicates that combining hybrid CNN with LSTM model with embedded features outperforms the CNN and LSTM models separately.

Although the dataset used has been used extensively in phishing research and analysis, it is still limited in sample size for certain types of attacks and rare cases, given the ever-evolving phishing techniques. Therefore the proposed model does not fully reflect the real situation in the everyday network environment. In the future we will focus on building models that can adaptively recognize new types of attack techniques that no previous model has seen. Another alternative takes advantage of the sophistication of large language models (LLMs).

Funding Statement

This work was supported by the Sun Moon University Research Grant of 2023.

References

[1] G. Tsochev, R. Trifonov, O. Nakov, S. Manolov, and G. Pavlova, "Cyber security: Threats and Challenges," *2020 Int. Conf. Autom. Informatics, ICAI 2020 – Proc.*, 2020, doi: 10.1109/ICAI50593.2020.9311369.

[2] A. K. Jain and B. B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterp. Inf. Syst.*, vol. 16, no. 4, pp. 527–565, 2022, doi: 10.1080/17517575.2021.1896786.

[3] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Eng. Appl. Artif. Intell.*, vol. 104, no. June, p. 104347, 2021, doi: 10.1016/j.engappai.2021.104347.

[4] A. Odeh, I. Keshta, and E. Abdelfattah, "Machine LearningTechniques-for Detection of Website Phishing: A Review for Promises and Challenges," *2021 IEEE 11th Annu. Comput. Commun. Work. Conf. CCWC 2021*, pp. 813–818, 2021, doi: 10.1109/CCWC51732.2021.9375997.

[5] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," *Mach. Learn. Knowl. Extr.*, vol. 3, no. 3, pp. 672–694, 2021, doi: 10.3390/make3030034.

[6] S. Maurya, H. S. Saini, and A. Jain, "Browser extension based hybrid anti-phishing framework using feature selection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 579–588, 2019, doi: 10.14569/IJACSA.2019.0101178.

[7] N. Zhao, "Credibility Evaluation of Web Big Data Information Based on Particle Swarm Optimization," *J. Web Eng.*, vol. 21, no. 2, pp. 405–423, 2021, doi: 10.13052/jwe1540-9589.21212.

[8] DBIR, "Data Breach Investigations Report (DBIR)," 2021. [Online]. Available: https://www.verizon.com/business/resources/reports/2021/2021-data-breach-investigations-report.pdf?_ga=2.226803477.929326497.1638808696-965423441.1638808696. [Accessed: 06-Aug-2023].

[9] TESSIAN, "Must-Know Phishing Statistics: Updated 2022," 2022. [Online]. Available: https://www.tessian.com/blog/phishing-statistics-2020/. [Accessed: 06-Aug-2023].

[10] D. Bera, O. Ogbanufe, and D. J. Kim, "Towards a thematic dimensional framework of online fraud: An exploration of fraudulent email attack tactics and intentions," *Decis. Support Syst.*, vol. 171, no. April, 2023, doi: 10.1016/j.dss.2023.113977.

[11] Anti-Phishing Work Group, "Phishing Activity Trends Report," *Phishing Act. Trends Rep.*, vol. Q2 2020, no. August, pp. 1–13, 2020.

[12] Wandera, "Mobile Threat Landscape 2020: Understanding THE Key Trend IN Mobile Enterprise Security IN 2020. Technical Report," 2020.

[13] A. Kumar, K. Abhishek, S. K. Shandilya, and D. M. Ghalib, "Malware Analysis Through Random Forest Approach," *J. Web Eng.*, vol. 19, 2020, doi: 10.13052/jwe1540-9589.195610.

[14] K. Raghunath, V. V. Kumar, M. Venkatesan, K. K. Singh, M. T R, and A. Singh, "XGBoost Regression Classifier (XRC) Model for Cyber Attack Detection and Classification Using Inception V4," *J. Web Eng.*, 2022, doi: 10.13052/jwe1540-9589.21413.

[15] F. Wan, F. Yang, T. Wu, D. Zhang, L. Zhang, and Y. Wang, "Chinese shallow semantic parsing based on multilevel linguistic clues," *J. Comput. Methods Sci. Eng.*, vol. 20, pp. 1–10, 2020, doi: 10.3233/JCM-194111.

[16] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput. Commun.*,

vol. 175, no. November 2020, pp. 47–57, 2021, doi: 10.1016/j.comcom .2021.04.023.

[17] K. A. Apoorva and S. Sangeetha, "Analysis of uniform resource locator using boosting algorithms for forensic purpose," *Comput. Commun.*, vol. 190, no. March, pp. 69–77, 2022, doi: 10.1016/j.comcom.2022. 04.002.

[18] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: detection of phishing websites by inspecting URLs," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 2, pp. 813–825, 2020, doi: 10.1007/s12652-019-01311-4.

[19] M. Sánchez-paniagua, E. Fidalgo, E. Alegre, and R. Alaiz-rodríguez, "Phishing websites detection using a novel multipurpose dataset and web technologies features," *Expert Syst. Appl.*, vol. 207, no. June, p. 118010, 2022, doi: 10.1016/j.eswa.2022.118010.

[20] S. W. Liew, N. F. M. Sani, M. T. Abdullah, R. Yaakob, and M. Y. Sharum, "An effective security alert mechanism for real-time phishing tweet detection on Twitter," *Comput. Secur.*, vol. 83, pp. 201–207, 2019, doi: 10.1016/j.cose.2019.02.004.

[21] M. Hussain, C. Cheng, R. Xu, and M. Afzal, "CNN-Fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet," *Inf. Sci. (Ny).*, vol. 631, no. July 2022, pp. 328–345, 2023, doi: 10.1016/j.ins.2023.02.039.

[22] F. Zheng, Q. Yan, V. C. M. Leung, F. Richard Yu, and Z. Ming, "HDP-CNN: Highway deep pyramid convolution neural network combining word-level and character-level representations for phishing website detection," *Comput. Secur.*, vol. 114, p. 102584, 2022, doi: 10.1016/ j.cose.2021.102584.

[23] E. Oram, P. B. Dash, B. Naik, J. Nayak, S. Vimal, and S. K. Nataraj, "Light gradient boosting machine-based phishing webpage detection model using phisher website features of mimic URLs," *Pattern Recognit. Lett.*, vol. 152, pp. 100–106, 2021, doi: 10.1016/j.patrec.2021.09.018.

[24] S. Mathulaprangsan, K. Lanthong, D. Jetpipattanapong, S. Sateanpattanakul, and S. Patarapuwadol, "Rice Diseases Recognition Using Effective Deep Learning Models," *2020 Jt. Int. Conf. Digit. Arts, Media Technol. with ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng. ECTI DAMT NCON 2020*, no. March, pp. 386–389, 2020, doi: 10.1109/ECTIDAMTNCON48261.2020.9090709.

[25] S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A. M. Al-Zoubi, and S. Kotti Padannayil, "Spam Emails Detection Based on Distributed Word Embedding with Deep Learning," *Stud. Comput. Intell.*,

vol. 919, no. January 2021, pp. 161–189, 2021, doi: 10.1007/978-3-030-57024-8_7.

[26] R. Wazirali, R. Ahmad, and A. A. K. Abu-Ein, "Sustaining accurate detection of phishing URLs using SDN and feature selection approaches," *Comput. Networks*, vol. 201, no. November, p. 108591, 2021, doi: 10.1016/j.comnet.2021.108591.

[27] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, 2021, doi: 10.1007/s11235-020-00733-2.

[28] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: A top-down survey," *Comput. Networks*, vol. 141, pp. 199–221, 2018, doi: 10.1016/j.comnet.2018.03.012.

[29] E. Elbasani and J. D. Kim, "AMR-CNN: Abstract Meaning Representation with Convolution Neural Network for Toxic Content Detection," *J. Web Eng.*, vol. 21, no. 3, pp. 677–692, 2022, doi: 10.13052/jwe1540-9589.2135.

[30] K. L. Chiew, C. L. Tan, K. S. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf. Sci. (Ny).*, vol. 484, pp. 153–166, 2019, doi: 10.1016/j.ins.2019.01.064.

[31] M. A. Adebowale, K. T. Lwin, E. Sánchez, and M. A. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text," *Expert Syst. Appl.*, vol. 115, no. December 2017, pp. 300–313, 2019, doi: 10.1016/j.eswa.2018.07.067.

[32] T. Mohammad, Rami, Thabtah, Fadi Abdeljaber and McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Comput. Appl.*, vol. 25(2), ISSN 0941-0643, pp. 443–458, 2014.

[33] F. A. Mohammad, Rami, McCluskey, T.L. and Thabtah, "Intelligent Rule based Phishing Websites Classification. IET Information Security," *IET Inf. Secur.*, vol. 8(3), ISSN 1751-8709, pp. 153–160, 2014.

[34] "PhishTank." [Online]. Available: https://phishtank.org/.

[35] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, pp. 345–357, 2019, doi: 10.1016/j.eswa.2018.09.029.

[36] Y. X. Y. Technologies, "yandex-xml," 2013. [Online]. Available: https://yandex.com.tr/dev/xml/.

[37] M. Alshehri, A. Abugabah, A. Algarni, and S. Almotairi, "Character-level word encoding deep learning model for combating cyber threats in phishing URL detection," *Comput. Electr. Eng.*, vol. 100, no. March, p. 107868, 2022, doi: 10.1016/j.compeleceng.2022.107868.

[38] X.-C. Z. P. Dong-Jie Liu, Guang-Gang Geng, "Multi-scale semantic deep fusion models for phishing website detection," *Econ. Lett.*, p. 110456, 2022, doi: 10.1016/j.eswa.2022.118305.

[39] A. Aljofey et al., "An effective detection approach for phishing websites using URL and HTML features," *Sci. Rep.*, vol. 12, no. 1, pp. 1–19, 2022, doi: 10.1038/s41598-022-10841-5.

[40] H. Wu, X. Zhang, and J. Yang, "Deep Learning-Based Encrypted Network Traffic Classification and Resource Allocation in SDN," *J. Web Eng.*, vol. 20, no. 8, pp. 2319–2334, 2021, doi: 10.13052/jwe1540-9589.2085.

[41] H. C. Altunay and Z. Albayrak, "A hybrid CNN + LSTMbased intrusion detection system for industrial IoT networks," *Eng. Sci. Technol. an Int. J.*, vol. 38, p. 101322, 2023, doi: 10.1016/j.jestch.2022.101322.

[42] W. El-Shafai, I. Almomani, and A. Alkhayer, "Visualized malware multi-classification framework using fine-tuned cnn-based transfer learning models," *Appl. Sci.*, vol. 11, no. 14, 2021, doi: 10.3390/app11146446.

[43] P. R. Kanna and P. Santhi, "Hybrid Intrusion Detection using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks," *Expert Syst. Appl.*, vol. 194, no. May 2021, 2022, doi: 10.1016/j.eswa.2022.116545.

[44] N. Gupta, V. Jindal, and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," *Comput. Networks*, vol. 192, no. December 2020, 2021, doi: 10.1016/j.comnet.2021.108076.

[45] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Syst. Appl.*, vol. 185, no. July, 2021, doi: 10.1016/j.eswa.2021.115524.

[46] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, Q. V. Pham, and N. N. Dao, "A LSTM-FCNN based multi-class intrusion detection using scalable framework," *Comput. Electr. Eng.*, vol. 99, no. December 2021, 2022, doi: 10.1016/j.compeleceng.2022.107720.

[47] M. Korkmaz, O. K. Sahingoz, and B. DIri, "Detection of Phishing Websites by Using Machine Learning-Based URL Analysis," *2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020*, 2020, doi: 10.1109/ICCCNT49239.2020.9225561.

Biographies

Candra Zonyfar received his bachelor's degree in computer science from Singaperbangsa Karawang in 2013 and his master's degree in computer science from Budi Luhur University, Indonesia in 2019. He is currently studying for a Ph.D. in computer and electronics engineering from Sun Moon University in 2022 South Korea. His main research interests include deep learning and data science in bioinformatics.

Jung-Been Lee received his bachelor's degree in computer engineering from Chosun University in 2002. He received his M.Sc. and Ph.D. degrees from the College of Informatics at Korea University, Seoul, in 2011 and 2020, respectively. From 2020 to 2022, he was a research professor at Chronobiology Institute at Korea University in Seoul. He is currently an assistant professor in the department of computer science and engineering at Sun Moon University, Asan, Korea. His primary areas of study include mining software artifacts and analysis and machine learning from wearable sensor data.

Jeong-Dong Kim received his bachelor's degree in computer engineering from Sun Moon University in 2005. He received his M.Sc. and Ph.D. degrees in Computer Science from Korea University at Korea in 2008 and 2012, respectively. He is an associate professor in the department of computer science and engineering, Sun Moon University, Asan, Korea. His research interests include bigdata analysis based on deep learning, healthcare, software and data engineering, and bioinformatics.