Deep Learning with Convolutional Neural Network and Long Short-Term Memory for Phishing Detection

*M. A. Adebowale*

*School of Computing and Information Science*

*Anglia Ruskin University*

*Chelmsford, UK*

[*moruf.adebowale@pgr.anglia.ac.uk*](mailto:moruf.adebowale@pgr.anglia.ac.uk)

*ORCID ID: 0000-0001-5704-4985*

*K. T. Lwin*

*School of Computing and Digital Technology*

*Teesside University,*

*Middlesbrough, UK*

[*k.lwin@tees.ac.uk*](mailto:k.lwin@tees.ac.uk)

*M. A. Hossain*

*School of Computing and Digital Technology*

*Teesside University,*

*Middlesbrough, UK*

[*a.hossain@tees.ac.uk*](mailto:a.hossain@tees.ac.uk)

***Abstract–Phishers sometimes exploit users’ trust of a known website’s appearance by using a similar page that looks like the legitimate site. In recent times, researchers have tried to identify and classify the issues that can contribute to the detection of phishing websites. This study focuses on design and development of a deep learning based phishing detection solution that leverages the Universal Resource Locator and website content such as images and frame elements. A Convolutional Neural Network (CNN) and the Long Short-Term Memory (LSTM) algorithm were used to build a classification model. The experimental results showed that the proposed model achieved an accuracy rate of 93.28%.***

***Keywords–Phishing detection; Cybercrime; Deep learning (DL); Convolutional Neural Network (CNN); Long Short-Term Memory (LSTM); Big data; Universal Resource Locator (URL).***

# Introduction

Nowadays, identity theft is one of the most popular cybercrime activities [1]. The most common method used to steal confidential information from online users is phishing. This activity is usually defined as a fraudulent attempt that is usually made via email and/or fake websites. Phishers are likely to have access to a wide variety of tactics and approaches that they can utilise to create a well-designed phishing attack. Cybercriminals use phishing for various illicit activities such as identity theft and fraud. They can also install malware on inadequately protected end-user systems to gain access to the systems of their victims [2]. Phishing is one of the critical threats to web activities, where the attacker mimics the website of an official establishment to gather the personal data of online users [3].

Several solutions have been proposed in the literature that use various methodologies to attempt to counter web-phishing threats. However, phishing is a complicated phenomenon to tackle as it differs from the other security threats such as intrusions and malware, which are based on the methodological security holes in network systems [4]. According to a report by the Anti-Phishing Working Group[[1]](#footnote-1), the number of phishing attacks detected in the first quarter of 2018 was 46% higher than in the fourth quarter of 2017. The most targeted sector is the payment service which is the object of 39.4% of phishing attacks, followed by software-as-a-service/webmail with 18.7%, financial institutions with 14.2% and other sectors with 16.4% [5].

Given the above, this research aims to develop a solution that includes the detection of phishing attacks as well as providing insights and improving awareness as to how active Internet users can protect themselves against phishing attacks. It is hoped that this research will help to formulate an upward trend in the practice of preventive measures against cyber-security issues.

Despite various approaches having been utilised to develop anti-phishing tools to combat phishing attacks, these methods suffer from limited accuracy. To address this issue, in this study we exploit the use of two essential, but previously under-studied method, the deep-learning (DL) algorithm that is categorised as a type of unsupervised machine learning algorithm that learns from the data on its own and designs a scheme for future use. This type of algorithm has a high probability of detecting newly generated phishing URLs and, moreover, does not need manual feature engineering. The use of a convolutional neural network (CNN) and Long Short-Term Memory (LSTM) proffers an effective solution for phishing website detection. Also, we explore and evaluate different CNNs and LSTM architectures that vary in terms of the width and depth of their layers to showcase the effect of varying the dataset spiral image context and scale performance. We then discuss when and why the transfer of learning from CNN and LSTM could be valuable.

The original contribution of this study lies in the ability of the proposed method of CNN and LSTM, which we named the Intelligent Phishing Detection System (IPDS), to use the image, frame and text content of a website to detect phishing activities by using a hybridised combination of the CNN and LSTM. To our best knowledge, this is the first work that considers the best-integrated text, image and frame feature based solution using deep learning algorithm (CNN+LSTM) for phishing detection scheme.

The proposed IPDS uses two DL layers to classify phishing websites by employing LSTM on text and frame content and the CNN on images. Thus, the model can easily explore the richness of the words embedded in the website’s Universal Resource Locator (URL) as well as the images on the site. The performance of the proposed model was tested by applying it to a phishing dataset that consisted of one million URLs taken from PhishTank and a legitimate site Common Crawl as well as over 10,000 images from legitimate and phishing websites that were personally collected from various e-commerce and e-banking sites over some time. This dataset was used to train and test the network using holdout cross-validation of 70% and 30% respectively. The results of our experiment showed that some level of improvement in phishing detection was achieved through the use of hybrid features by combining image, text and frame of a site with the use of the DL algorithm. Furthermore, in our experiment, we obtained information about the usefulness of unsupervised pre-training and the effectiveness of image feature extraction in detecting phishing sites.

The rest of this paper is arranged as follows: Section II contains the literature review. Section III presents the methodology, including the CNN, LSTM, and DL algorithm. Section IV describes the experiment. The section presents the results and analysis. Section VI contains the conclusion and directions for future work.

# Literature review

In recent times, artificial intelligence technology has come to drive many aspects of modern society, from social networking and web searching to content filtering and e-commerce. It is also present in consumer products such as cameras and smartphones. Moreover, machine learning techniques are used for object identification in images, the transcription of voice into text, matching news items and products with users’ interests and presenting relevant search results.

The DL concept started with the study of artificial neural networks (ANNs). This concept has become an active research area in recent years. To build a standard neural network (NN) requires the use of neurons to produce real-valued activations, and with the adjustment of weights, the ANN behaves as required [6]. However, training the ANN with backpropagation make it useful, as the gradient descent algorithms which have played a vital role in the model in the past decades. However, while training accuracy can be high using this approach, the performance of backpropagation when applied to the testing data might not be satisfactory [7]. James, Sandhya and Thomas [8], propose a scheme for detecting phishing websites based on their features and the URL of the site using machine learning techniques. Also, they discuss the techniques used for phishing detection based on the lexical features, page properties and host features. They evaluate different data mining algorithms for understanding the structure of URLs that spread phishing contents. They have input features save in comma separated value (.csv) with use of four machine learning algorithms considered for processing the features. The machine learning algorithm used are Naïve Bayes, J48 decision tree, K-nearest neighbour (KNN) and support vector machine (SVM). The result of the experiment shows that the J48 decision tree with a lexical feature performs better the rest of the algorithm with an accuracy of 93.2%. The approach to their solution is better but require an additional feature to make it more robust [8].

Le *et al.* [9]proposed a solution named URLNet, which is an end-to-end deep learning framework to learn a nonlinear malicious URL by detecting from the URL. They applied a convolutional neural network to both words and characters of URL features to learn the URL embedding in a jointly optimised framework. This approach allows their model to capture several types of semantic data that was not possible by existing schemes. Also, they present an advanced word-embeddings to solve the problem of too many rare words observed in the task. A large dataset was used to perform their experiments on a large-scale dataset and demonstrate a strong performance over the current method. The approach has two branches; the first branch has a character level CN, where the character embedding is used to represent the URL. Hence, the second branch contains the word-level CNN, where a word-level embedding is used to represent the URL. However, the word embedding itself is a mixture of the character-level embedding and the individual words embedding of that word. Their approach worked in a manner that does not require any expert features [9].

Likewise, Yi *et al.* [10], design two sets of features for web phishing interaction element and original features. Also, they develop a scheme based on a deep belief network (DBN) is then presented. The test includes using the real Internet protocol (IP) flows from the ISP (Internet Service Provider) reflect that the detecting model based on DBN can achieve an approximately 90% true positive rate [10].

The procedure for training a CNN using backpropagation follows the same process as that for a normal NN. However, Yoshua [11] proposed an alternative method for training CNN called the error gradient and applied the CNN to classify images. In the initial stage of that method, the information is spread in a feed-forward direction through various layers. During this stage, a digital filter is applied to obtain important features at each layer to compute the value of the output. Also, in the next step, the error among the predictable and real values of the output is computed. Unlike the standard DL algorithm for backpropagation and error minimisation, in the CNN, the weight matrix is adjusted, and the network is fine-tuned to improve image classification and reduce pre-processing. However, the parameter setting is not required in the CNN, unlike in the traditional NN; the error gradient method trains the filters in the CNN which are independent of prior knowledge and human interference [11].

In the aspect of feature extraction, CNN has also provided a solution. There is a solution proposed in [12] on the method that combines difficulty with an encoder-decoder architecture. In their method, sparse predictive disintegration unsupervised feature learning is used with the sparsity limitations on the element direction that is based on an encoder-decoder architecture. Their feature extraction stage involves a filter bank, a non-linear transformation, and a feature pooling layer [12]. Mathieu, Henaff and LeCun [13] proposed an algorithm which fast-track training and interference by a critical factor, which yields enhancements performance over a magnitude compared to the existing advanced application [13].

On the other hand, Bahnsen *et al.*, [14] explored the use of LSTM with features from URLs as input for machine learning schemes that could be applied for phishing site prediction. They use LSTM units to build a model that receives a URL in the form of a feature sequence as input to predict whether a website link goes to a phishing or legitimate site. In the model, each input feature is translated by a 128-dimension embedded and fed into the LSTM layer as a 150-step sequence. Hence, classification is done using an output signed neuron [14].

# Methodology

In this study, we use both the LSTM layer and the CNN to extract features from different websites, where the output is a binary number that reflects whether or not the input sequences are real or fake (phishing).

## Convolutional Neural Network

The CNN is a type of architecture that is discriminative and shows satisfactory performance in processing two-dimensional data with grid topologies, such as images and videos. The concept of CNN is superior to that of NN in terms of time-delay. In the CNN concept, the weights are shared in a temporal dimension, which leads to a decrease in computation time. The general matrix multiplication in the standard NN is therefore replaced in the CNN. Hence the CNN approach reduces the weights, thereby decreasing the complexity of the network. Consequently, the feature extraction procedure in a standard learning algorithm can be enhanced by directly importing images into the network as raw inputs. This type of model for the training of the architecture layers led to the success of the first DL algorithms.

Furthermore, the use of the standard backpropagation algorithm enables CNN topology to influence three-dimensional connections to decrease the number of parameters in the network and improve its performance. Another benefit of the CNN model is the lower pre-processing requirement. The use of the Graphics Processing Unit (GPU) has accelerated computing techniques and has been exploited to develop the computational requirements of CNN rapidly. Hence, in recent times, CNN's have been applied to image classification, face detection, speech recognition, handwriting recognition, behavioural recognition and recommender systems.

### CNN Algorithm structure

There are three main components in the learning process of a CNN: equal representation, sparse interaction and parameter sharing [15]. The CNN is different from the standard NN, which draws out the connection among the input and output units from matrix development. In contrast, CNN decreases the computational load with a thin interface where the kernels are made slighter than inputs and are used for the entire image. Also, in the CNN, the idea behind parameter allocation is that, as a substitute of learning a detached set of parameters at each location, the CNN only needs to learn a set of features, which allows the CNN to perform better than NN. Also, the CNN has a beautiful property called equivariance, that works with parameter distribution so that every time the input changes the output follow suits. Hence, CNN requires fewer parameters than legacy NN algorithms. This requirement leads to a reduction in memory usage and improves efficiency.

The components of the standard CNN layers are illustrated in Fig. 1. The figure shows how the input image is convolved with trainable filters with possible offsets to produce features maps in the first c-layer. The filters contain a layer of connection weights. In a real sense, there four pixels in the feature map from a group. These pixels pass through a sigmoid function to produce additional feature maps in the first s-layer. This process is continued to obtain the feature maps in the following c-layers and s-layers. Then, at the end of this process, the values of these pixels are rasterised and displayed in a single vector as the input of the network [16].



Fig. 1: Schematic Structure of Fully Connected Convolutional Neural Network (Source: LeNet)

.

The layer that is responsible for the collection of elements when the input of each neuron is connected from the previous layer is called the c-layer. After the local features are extracted, the positional association can then be identified. Also, the function kernel has a slight influent over the activation function used by the sigmoid function to achieve scale invariance. Hence, the model uses the filter to connect the series of overlapping available fields and convert the 2D image set from the input to a single element in the output. However, when overfitting occurs, a pooling process called sub-sampling is presented to decrease the total size of the signal. This solution has been used for data size reduction in audio compression [13].

## Deep Long Short-Term Memory

In this study, we also use LSTM as an algorithm that forms part of the structure of our scheme that takes the input from a URL as a character sequence and predicts whether the link is a phishing or legitimate website. Long Short-term Memory is an adaptive recurrent neural network (RNN), where each neuron is swapped by a memory cell which is additional to the conservative neuron on behalf of an internal state. It also uses multiplicative units as gates to control the flow of information. The LSTM layers consist of a set of repeatedly linked blocks called memory blocks. These blocks each contain one or more recurrently connected memory cells. Hence, a normal LSTM cell has an input gate that controls the input of data from outside the cell, which determines whether the cell keeps or overlooks the data in the internal state and an output gate that prevents or allows the inner state to be seen from the outside [14].

Furthermore, LSTM units are known to have the ability to learn extensive range dependency from input sequences. The LSTM training algorithm uses an error gradient for its calculation, where it combines real-time recurrent learning and backpropagation [17]. However, backpropagation is dropped after the first timestamp because the long-term dependencies are dealt with by the memory blocks, and not by the flow of the backpropagation error gradient. This step has further helped in making LSTM directly comparable to other RNNs in terms of performance because training can be done with standard backpropagation with time [18].

### LSTM Algorithm architecture

The central components of the LSTM architecture are the memory cell, which can maintain its state over some time and non-linear gate units, which regulate the information input and output flow of the network [19]. Based on the insights derived from secure networks, it is considered that because the LSTM neuron consists of internal cells and gate units, one should not only look at the output of the neuron but also at the internal structure to design original features for LSTM so that it can address classification problems [20].

Figure 2 shows the architecture of an LSTM in which there are three bidirectional LSTM layers, two feed-forward layers, and a SoftMax layer that gives the predictions. This fully connected architecture allows us to take advantage of the inherent correlations among connections. Before the second layer in the network, co-occurrence exploration is applied to the connection to learn from the input features. Lastly, backpropagation is applied to the LSTM layer to allow more effective learning [21].

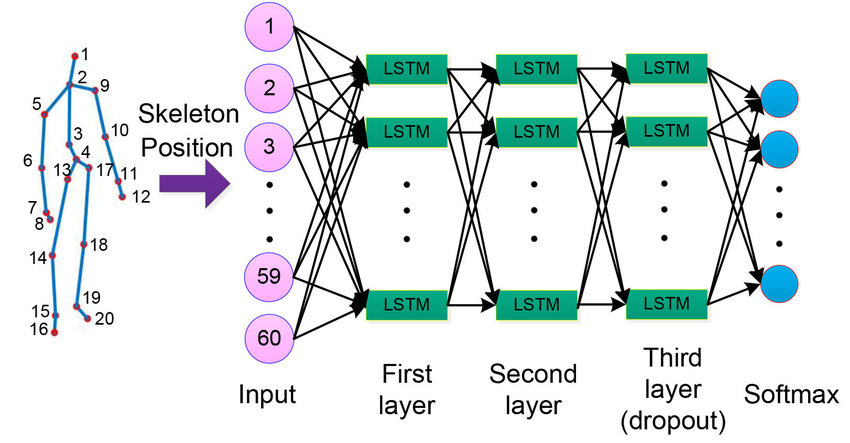


Fig. 2: Schematic Structure of Fully Connected Long Short-Term Memory (Source: Zhu et al., 2016)

In this study, CNN and LSTM are used to build a hybrid model, the IPDS, which can be used to classify phishing websites. The general structure of the IPDS is presented in Fig. 3.

The aim behind this conceptualisation is to integrate the CNN, LSTM and a DL algorithm and apply them to the features extracted from websites to detect phishing activities more accurately. Based on a comparison between the features extracted and the knowledge model, the classification of legitimate and phishing sites is achieved. Websites are evaluated individually to determine whether they are legitimate or spoof sites. In the proposed method, the features of the web page that have similarity with the proposed solution are compared to remove duplication in the feature set. Then the feature set is used to train the model that is used in the classification process.

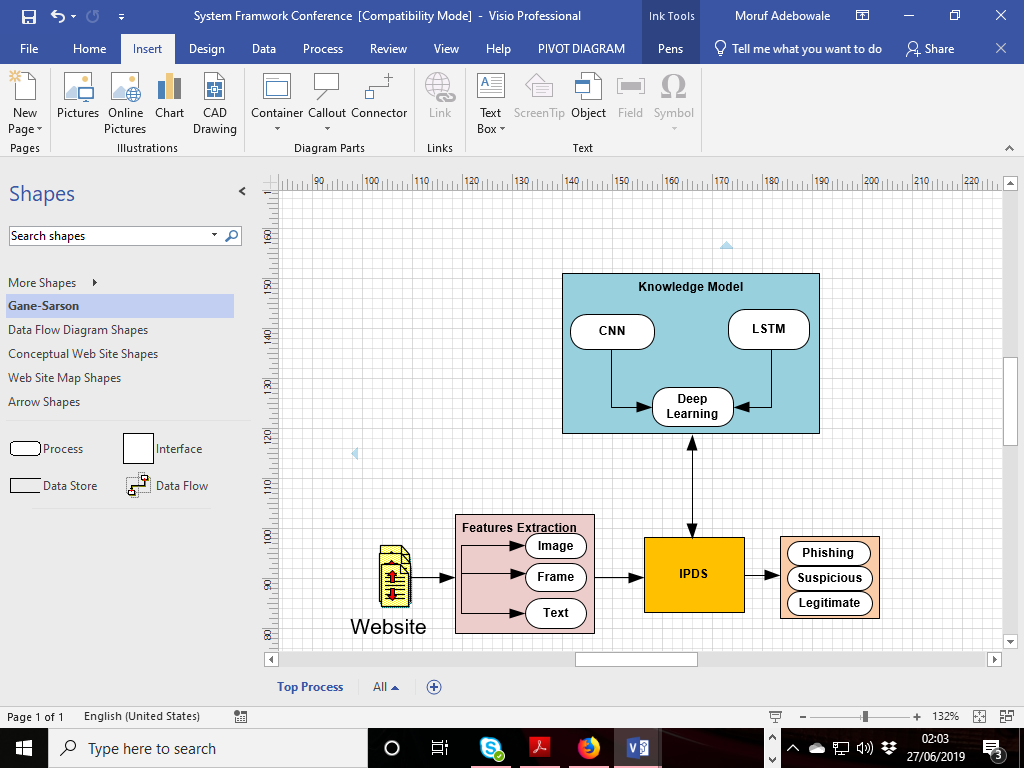


Fig. 3: Intelligent Phishing Detection System (IPDS) Structure

The overall conceptual framework for the intelligent phishing detection system (IPDS) structure using deep learning is presented in Fig. 3. The concept involves using two deep learning algorithms, namely LSTM and CNN, on different types of features that have been extracted from websites in order to better predict phishing activities. The feature extraction step and machine learning are applied in the initial stage in our classification process. The structure diagram of the IPDS in Fig. 3 above illustrates the process of acquiring the website features and feeding them into the deep learning system for classification purposes. Then, the trained LSTM-CNN network is applied to distinguish accurately between legitimate, suspicious and phishing websites in real-time.

Based on the differences between the features extracted and the IPDS model, the classification of websites as either legitimate, suspicious or phishing is achieved. Websites are assessed separately to ascertain whether they are legitimate or fake (phishing). The feature set is used in the classification process, where 70% is used for the training set, and 30% is used for the testing set.

# Experiment setup

To train both the LSTM and CNN, a dataset was constructed that consisted of legitimate and phishing URLs. In total, 1 million URLs were used to train LSTM. Half of the dataset consisted of phishing sites from PhishTank, which is a site that is used as phishing URL depository, and half of the dataset was comprised of legitimate sites from Common Crawl, a corpus of web crawl data. To train the CNN, we collected more than 10,000 images from both legitimate and phishing sites. This dataset was used to train and test the network using holdout cross-validation of 70% and 30% respectively

## Data Pre-processing

The raw data from both images and URLs contained a lot of background information and varied in length and sizes. Therefore, we had to pre-process this data to make it available for training the model. For the CNN architecture, we cropped images from the sites based on the springing box and merely removed the wrong image. For the LSTM architecture, we collect several URLs and save them in Microsoft Excel as comma-separated values with only the URL in one column and their category label in the other column (Table 2).

## Model Details

The model was developed in Matlab version 9.5 using the deep learning toolbox. For the CNN architecture, there were three categories of data. The image (Fig. 4) is a sample of image which was loaded into the image data store and processed to extract the speeded-up robust features (SURF) from all images using the grid method to create a bag of features where the GridStep was set to [8 8] with BlockWidth of [32 64 96 128]. Then we used clustering to create a 1000-word visual vocabulary. For the LSTM architecture, the dataset was partitioned, and holdout cross-validation was set to 0.3 for training and validation. The URLs were tokenised to separate each URL into a series of separate words, all of which were set in lowercase. The tokenised data was then encoding to make it available for training, where the maximum length was set to 75, the hidden size was set to 180, and the embedding dimension was 100 with the fully connected network. The training options were set to *adam*; epoch = 100, gradient threshold = 1, learning rate = 0.01 and verbose = false. By doing this, we were able to tweak the network architecture layer that included the parameter mentioned above to achieve better training accuracy.

# Results

The evaluation of the proposed method was performed based on traditional feature engineering, plus the classification algorithm methodology presented in Section III. We created features based on the URLs and images features and website element. We trained the CNN and LSTM classifier using one million URLs and over 10,000 images to build our model.

For the experimental results, we performed three series of experiments for each evaluation method, testing them against legitimate, suspicious and phishing websites. In the time-based evaluation process, time-stop at the point the scheme was able to classify the against all the legitimate datasets, suspicious datasets and phishing datasets. Then the process was also repeated severally to determine the average time to each classification at an interval. In the accuracy-based assessment, all the legitimate datasets, suspicious datasets and phishing datasets were utilised to test the toolbar.

We tested the accuracy of the model using the holdout cross-validation strategy. In the experiment, the overall classification accuracy result (Chart 1) for the proposed IPDS (CNN+LSTM) was 93.28% (Table 1). The best relative performance in classification was achieved by CNN with 92.55% and that for testing was achieved by LSTM with 92.79% (Table 1). Thus, the results showed that the accuracy of the proposed model was at 93.28%.

The results of our experiment showed that some level of improvement in phishing detection was achieved through the use of hybrid features by combining the images, text and frames of a site with the use of a hybrid DL algorithm. Furthermore, in our experiment, we obtained information about the usefulness of unsupervised pre-training and the effectiveness of image feature extraction in detecting phishing sites.

Table 1: Relative performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Accuracy % | Recall % | Precision % | | F-measure % |
| CNN | 92.55 | 92.51 | 92.58 | 92.54 | |
| LSTM | 92.79 | 92.78 | 92.81 | 92.80 | |
| IPDS | 93.28 | 93.27 | 93.30 | 93.29 | |

Chart 1: Experimental result for CNN, LSTM and (CNN+LSTM) IPDS classification

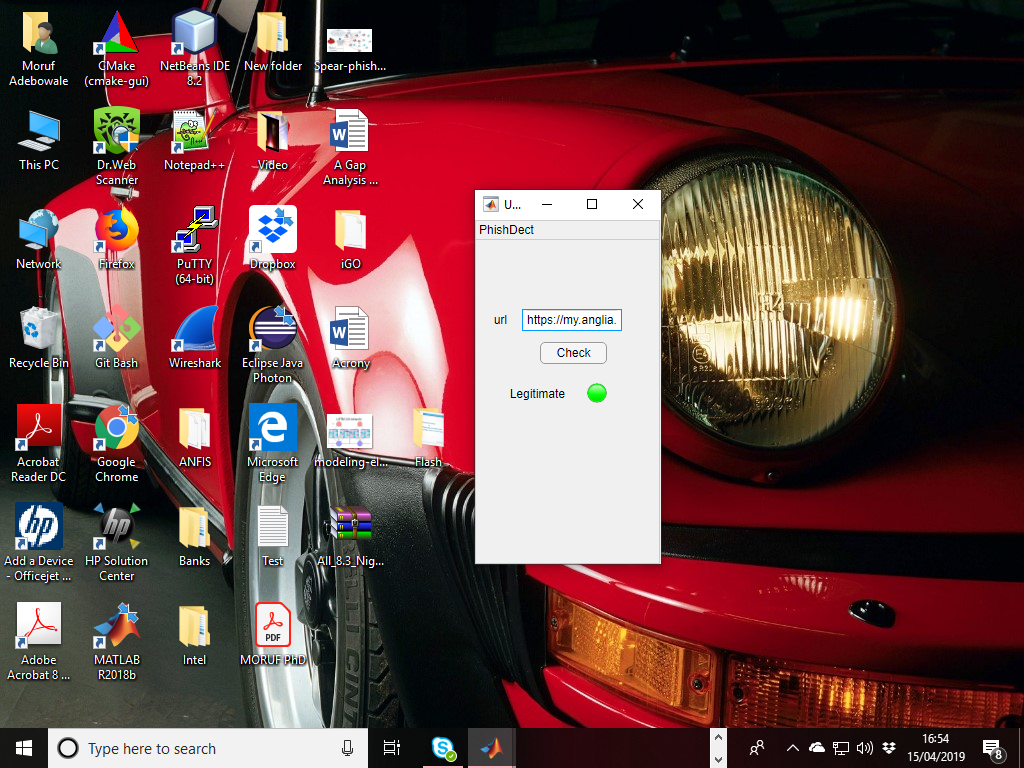


Fig. 4: Legitimate URL Check with Application Interface

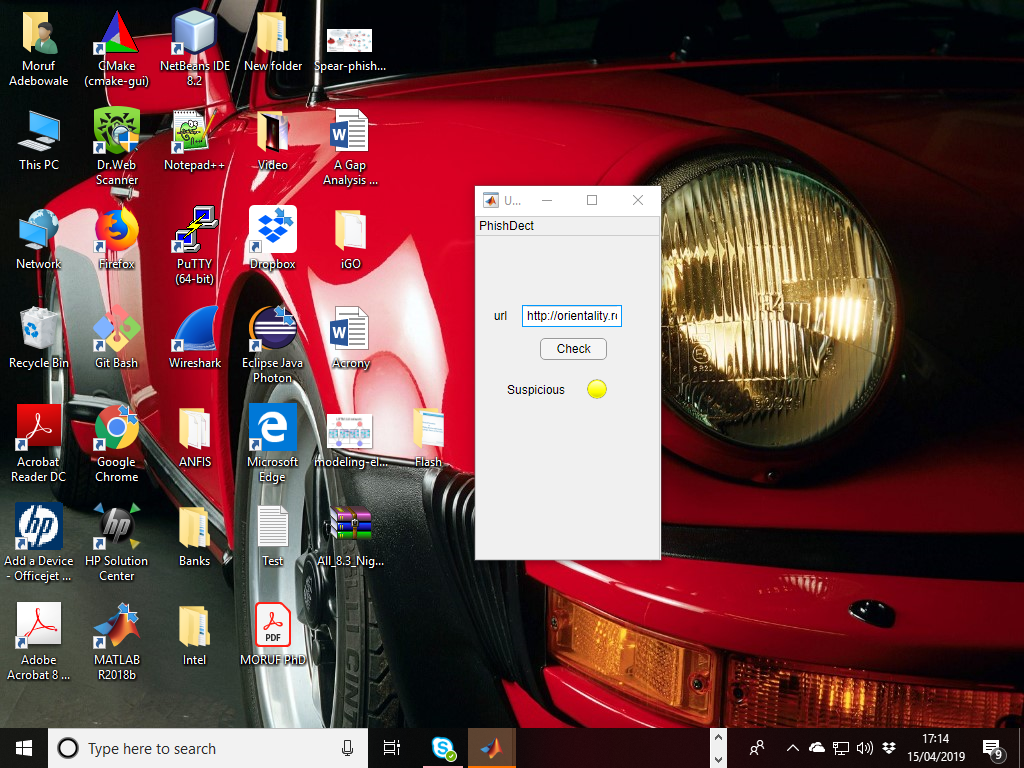


Fig. 5: Suspicious URL Check with Application Interface

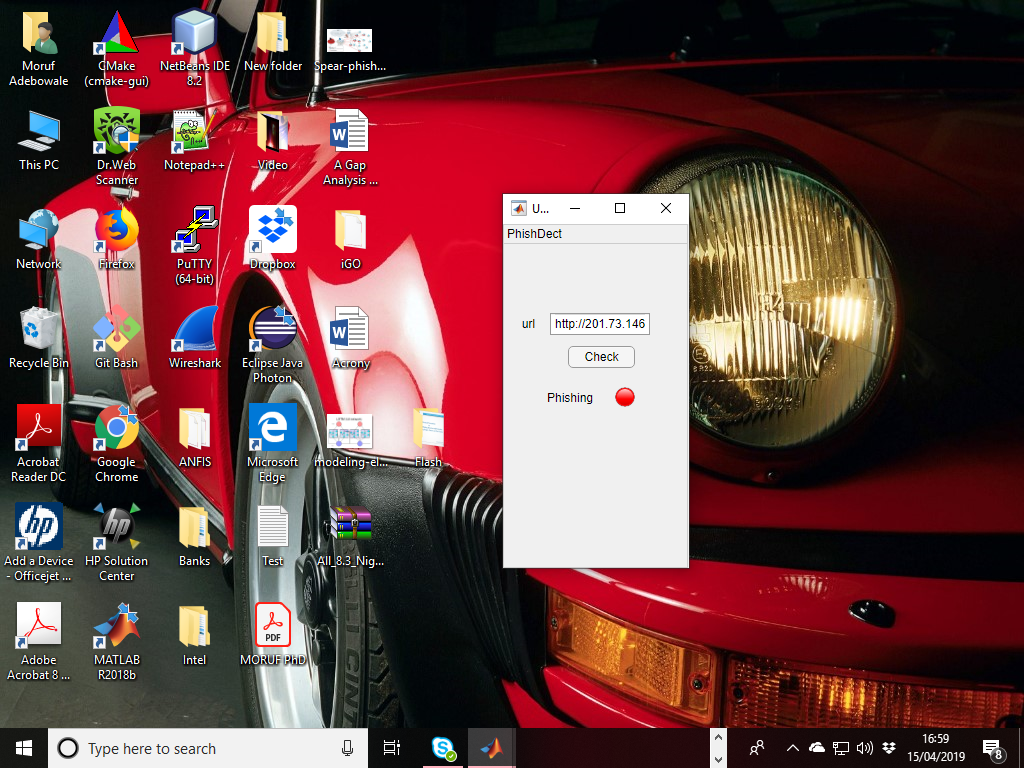


Fig. 6: Phishing URL Check with Application Interface

Table 2: Relationship between URL and Various Category Labels

|  |  |
| --- | --- |
| **URLs** | **Category** |
| http://bcpzornaseguras.com/bbvacontinentalpe-enlinea-20938209d23kjdh23d90238d23jwxj23/ | Phishing |
| https://www.google.com/ | Legitimate |
| https://www.microsoft.com/en-gb/ | Legitimate |
| http://kelberdesigner.com/adesao/eng/2.html | Phishing |
| http://www.stopagingnews.com/wp-admin/js/wells/ibrowellsup/identity.php | Phishing |
| http://orientality.ro/RENNE/ourtime.com/ourtime.com/ourtime.html | Suspicious |
| http://bcpzornaseguras.com/ | Suspicious |
| http://www.vinaros.org/locale/es/fb/ | Phishing |
| http://www.vinaros.es/locale/es/fb/ | Phishing |
| http://bcpzonasegura.viai1bcp.com/bcp/0peracionesnlinea/ | Phishing |
| http://riquichichichi.tk/ptm/web/ | Phishing |
| http://unitedstatesreferral.com/santos/gucci2014/gdocs/gucci.php?Acirc=A?A?=A?Auffe0= | Phishing |
| http://www.Legitimategovbr.com/SIIBC/ | Phishing |
| http://201.73.146.167/teste/ | Phishing |
| https://my.anglia.ac.uk/CookieAuth.dll?GetLogon?curl=Z2F&reason=0&formdir=3 | Legitimate |
| https://uk.yahoo.com/?p=us | Legitimate |



Fig. 7: Sample of Images Used to Train the CNN

# Conclusion and Future work

This study explored the possibility of differentiating unique legitimate URLs from phishing URLs by using two techniques, the CNN and LSTM, as a combined classifier in a novel approach called the Intelligent Phishing Detection System (IPDS). To evaluate the proposed hybrid approach, we used a dataset containing one million legitimate and phishing URLs from both PhishTank and Common Crawl dataset, as well as 10,000 images that were personally collected from both phishing and legitimate websites. The proposed IPDS gave an excellent classification accuracy of 93.28%.

Based on this result, we can deduce that distinguishing websites by their URLs, similarity context, and the images on the site by their pattern is an effective way of detecting phishing websites. The IPDS was able to respond with great agility and could verify a URL in 30.5 seconds.

In addition, our analysis revealed the advantages and disadvantages of both the CNN and LSTM methods. On the one hand, LSTM has an overall higher prediction performance, but there is a need for expert knowledge when creating the features. On the other hand, the CNN perform better, but achieves a performance that is, on average, a bit lower than that of LSTM. Hence, combining the two methods leads to a better result with less training time for LSTM architecture than the CNN model.

We carried out the extensive experimental analysis of the proposed hybrid approach in order to evaluate its effectiveness in the detection of phishing web pages and phishing attacks on large datasets. Then the sensitiveness of the proposed approach to various factors such as the type of features, number misclassification and split issues was studied. The result of our experiments showed that the proposed approach was highly effective in the detection of website phishing attacks as well as in the identification of fake websites. To the best of our knowledge, this is the first work that considers how best to integrate image, text and frame features into a combined solution for a phishing detection scheme.

Future work will include developing a web browser plug-in based on a DL algorithm to detect web phishing and thus protect users in real-time.

**References**

[1] L. De Kimpe, M. Walrave, W. Hardyns, L. Pauwels, and K. Ponnet, "You have got mail! Explaining individual differences in becoming a phishing target," *Telematics and Informatics,* vol. 35, no. 5, pp. 1277-1287, 01 August 2018, 2018, doi: <https://doi.org/10.1016/j.tele.2018.02.009>.

[2] H. Thakur and S. Kaur, "A Survey Paper On Phishing Detection," *International Journal of Advanced Research in Computer Science,* vol. 7, no. 4, pp. 64-68, 2016.

[3] M. Imani and G. A. Montazer, "Phishing website detection using weighted feature line embedding," *The ISC International Journal of Information Security,* vol. 9, no. 2, pp. 49-61, 2017.

[4] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems,* vol. 94, no. 2019, pp. 27-39, 01 May 2019 2019, doi: <https://doi.org/10.1016/j.future.2018.11.004>.

[5] APWG, "Unifying the Global Response to Cybercrime," in "Phishing Activities Trends Report, Third Quarter 2018," Anti-Phishing Working Group, Washington, D.C, USA, Online 31 July 2018 2018. [Online]. Available: <http://docs.apwg.org/reports/apwg_trends_report_q1_2018.pdf>

[6] P. Yang, G. Zhao, and P. Zeng, "Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning," *IEEE Access,* vol. 7, pp. 15196-15209, 11 January 2019 2019, doi: 10.1109/ACCESS.2019.2892066.

[7] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing,* vol. 234, pp. 11-26, 2017/04/19/ 2017, doi: <https://doi.org/10.1016/j.neucom.2016.12.038>.

[8] J. James, L. Sandhya, and C. Thomas, "Detection of phishing URLs using machine learning techniques," in *2013 International Conference on Control Communication and Computing (ICCC)*, Thiruvananthapuram, India, 13-15 Dec. 2013 2013: IEEE, pp. 304-309, doi: 10.1109/ICCC.2013.6731669.

[9] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "URLnet: Learning a URL representation with deep learning for malicious URL detection," presented at the arXiv preprint arXiv:1802.03162, Washington, DC, US, 2 March 2018, 2018.

[10] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, and T. Zhu, "Web phishing detection using a deep learning framework," *Wireless Communications and Mobile Computing,* vol. 2018, pp. 1-9, 2018.

[11] B. Yoshua, M. Jordan, Ed. *Learning Deep Architectures for AI* (Learning Deep Architectures for AI, no. 1). Hanover, MA USA: now Publishers Inc, 2009, pp. 1-127.

[12] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *12th International Conference on Computer Vision*, Kyoto, Japan, 29 Sept.-2 Oct. 2009 2009: IEEE, pp. 2146-2153, doi: 10.1109/ICCV.2009.5459469.

[13] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through ffts," *arXiv preprint arXiv:1312.5851,* pp. 1-9, 2013.

[14] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing URLs using recurrent neural networks," presented at the APWG Symposium on Electronic Crime Research (eCrime), Scottsdale, AZ, USA, 25-27 April 2017, 2017.

[15] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning. Book in preparation for MIT Press," *URL¡* [*http://www*](http://www)*.deeplearningbook.org,* 2016.

[16] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," *IEEE Computational Intelligence Magazine,* vol. 5, no. 4, pp. 13-18, 2010, doi: 10.1109/MCI.2010.938364.

[17] Z. Xu, S. Li, and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia, 3-6 November 2015 2015: IEEE, pp. 141-145.

[18] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks,* vol. 18, no. 5-6, pp. 602-610, 2005.

[19] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE transactions on neural networks and learning systems,* vol. 28, no. 10, pp. 2222-2232, 2017.

[20] D. Hakkani-Tür *et al.*, "Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM," in *Interspeech*, San Francisco, USA, 8–12 September 2016 2016, pp. 715-719.

[21] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA, 12–17 February 2016 2016: AAAI, pp. 3697-3703.

1. APWG report [Online] Available at: <https://docs.apwg.org/reports/apwg_trends_report_q4_2017.pdf> [Accessed 31 March 2018] [↑](#footnote-ref-1)