Intelligent Phishing Detection Scheme Using Deep Learning Algorithms

M. A. Adebowale
School of Computing and Information Science
Anglia Ruskin University
Chelmsford, UK
moruf.adebowale@pgr.anglia.ac.uk

K. T. Lwin
School of Computing and Digital Technology,
Teesside University,
Middlesbrough, UK
k.lwin@tees.ac.uk

M. A. Hossain
School of Computing and Digital Technology
Teesside University,
Middlesbrough, UK
a.hossain@tees.ac.uk

Abstract

Purpose – Phishing attacks have evolved in recent years due to high-tech-enabled economic growth worldwide. The rise in all types of fraud loss in 2019 has been attributed to the increase in deception scams and impersonation, as well as to sophisticated online attacks such as phishing. The global impact of phishing attacks will continue to intensify and thus a more efficient phishing detection method is required to protect online user activities. To address this need, this study focused on the design and development of a deep learning-based phishing detection solution that leveraged the universal resource locator and website content such as images, text and frames.

Design/Methodology/Approach – Deep learning techniques are efficient for natural language and image classification. In this study, the convolutional neural network (CNN) and the long short-term memory (LSTM) algorithm were used to build a hybrid classification model named the Intelligent Phishing Detection System (IPDS). To build the proposed model, the CNN and LSTM classifier were trained by using one million universal resource locators and over 10,000 images. Then, the sensitivity of the proposed model was determined by considering various factors such as the type of feature, number of misclassifications and split issues.

Findings – An extensive experimental analysis was conducted to evaluate and compare the effectiveness of the IPDS in detecting phishing web pages and phishing attacks when applied to large datasets. The results showed that the model achieved an accuracy rate of 93.28% and an average detection time of 25 seconds.

Originality/value – The hybrid approach using deep learning algorithm of both the CNN and LSTM methods which was used in this research work. On the one hand, the combination of both CNN and LSTM was used to resolve the problem of a large dataset and higher classifier prediction performance. Hence, combining the two methods leads to a better result with less training time for LSTM and CNN architecture, while using the image, frame and text features as a hybrid for our model detection. The hybrid features and IPDS classifier for phishing detection was the novelty of this study to the best of the authors' knowledge.

Keywords – Phishing detection; Cybercrime; Deep learning; Convolutional neural network (CNN); Long short-term memory (LSTM); Big data; universal resource locator (URL).

Paper type: Applied Artificial Intelligence (AI) research paper

1 Introduction

Phishing is a fraudulent online technique that is used to acquire sensitive and confidential user information (Jaison and Francis, 2014). The upsurge in phishing attacks is causing the threat of online identity theft to rise and these attacks often result in substantial monetary loss (Aggarwal, Rajadesingan and Kumaraguru, 2012). These circumstances have made the security of commercial transactions on the Internet less safe (Zuhair and Selamat, 2017). The Internet has become an effective means of communication, with many businesses using it to generate an online environment to manage offline commercial activities (Arachchilage and Love, 2014). However, even when the Internet is used to set up a solely online business functionality, despite the benefit that the Internet offers there is also a negative aspect that requires that the user pay attention to issues such as identity theft, fraud, malware and phishing attacks (Ghafir *et al.*, 2018).

Criminals deceive users into providing their confidential information that can then be used for identity theft (Arachchilage, Love and Beznosov, 2016). According to a report by the Anti-Phishing Working Group (APWG)[1], the number of phishing attacks discovered in the second quarter of 2019 was up 36% over the fourth quarter of 2018, while the most targeted sector is software-as-a-service (SaaS)/webmail which accounted for 36% of phishing attacks over the same period, followed by the payment service sector with 22%, financial institutions with 18% and other sectors with 9% (APWG, 2019). A phishing attack may appear in various forms of communication such as messages, voice over Internet protocol, short message service and spam emails (Ahmed and Abdullah, 2016). However, phishing attacks are mainly delivered via an email that lures users to click a link in the body of the email that then takes them to an external website that targets their financial information by claiming to be their bank, the Inland Revenue, a utility company or a government agency (Office for National Statistics, 2017). Criminal gangs also use malware and phishing emails as a means to compromise customers' details and security (Shaikh, Shabut and Hossain, 2016).

Given the above, security professionals are seeking to diminish the impact of phishing by using filtering techniques to identify spam and phishing emails, and also educating users and encouraging the use of anti-phishing toolbars that are designed to prevent users from accessing phishing web pages where their sensitive information would be requested and then transmitted to criminals (Bullee *et al.*, 2017). Thus far, various approaches have been utilised to develop anti-phishing tools to combat phishing attacks; however, they suffer from limited accuracy (Chorghe and Shekokar, 2016). Therefore, this research aimed to develop a solution that would enable the more accurate and timely detection of phishing attacks and also improve the awareness of active Internet users as to how they can protect themselves against phishing attacks.

[1] APWG report [online]. Available at: https://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf [Accessed 31 October 2019].

To date, several solutions, using various methodologies, have been proposed to counter the upward trend in phishing attacks (Hawanna, Kulkarni and Rane, 2016). A number of these solutions have used the deep learning algorithm to address this problem. This algorithm is categorised as a type of unsupervised machine learning algorithm, i.e., it learns from existing data on its own and then applies its knowledge to new data. Hence this type of algorithm has high potential in terms of being able to detect newly generated phishing websites. Moreover, it does not require the use manual feature engineering. In this study, the advantages of deep learning that are offered by the long short-term memory (LSTM) algorithm and the convolutional neural network (CNN) were used in combination to develop an effective solution for phishing website detection. We also explored and evaluated the differences in the LSTM and CNN architectures, which vary in terms of the width and depth of their layers, in order to showcase their effect on the dataset spiral image context and scale performance.

The original contribution of this study lies in the improved detection ability of the proposed hybrid LSTM and CNN method, which we named the Intelligent Phishing Detection System (IPDS). The proposed IPDS uses the image, frame and text content of a web page to detect phishing activities by using a hybridised combination of the LSTM algorithm and the CNN. This hybridisation of a deep learning algorithm (LSTM+CNN) is an extension of our previous work (Adebowale *et al.*, 2019) that sought to identify best-integrated text, image and frame features for use in a phishing detection solution. The previous work used the adaptive-network-based fuzzy inference system algorithm to classify phishing websites, which took an average of 30 seconds to classify phishing websites. In contrast, the IPDS proposed in the current study takes less than 25 seconds. The improvement on the previous work is due mainly to the deep knowledge base built with the LSTM+CNN algorithm that is capable of detecting newly generated phishing websites and does not need manual feature engineering to detect phishing sites.

The current work is timely because it has become necessary to develop a phishing detection system that is more efficient and accurate in terms of its decision-making ability. There is an urgent need for a robust system that can gather and analyse data as well as communicate with other systems efficiently to learn from experience and adapt according to current data in order to accurately monitor and identify phishing web pages (Barraclough *et al.*, 2013). The proposed IPDS uses two deep learning layers to classify phishing websites by employing the LSTM algorithm to assess text and frame content and the CNN to check images. A large dataset was collated and divided into two parts (70% and 30%, respectively) in order to train and test the CNN by using holdout cross-validation. The results of our extensive experiments showed that some level of improvement in phishing detection was achieved through the use of a deep learning algorithm on the image, text and frame features of websites.

The rest of this paper is arranged as follows: Section 2 contains the literature review. Section 3 presents the methodology, including the CNN, LSTM, and deep learning algorithm. Section 4 describes the experiments. Section 5 presents the results and analysis. Section 6 contains the conclusion and directions for future work.

2 Related Work

This section presents a review of the existing methods, tools and techniques that have been used for phishing detection. Currently, machine learning is demonstrating its effectiveness in an extensive range of applications. This technology has come to the forefront in recent times, owing to the advent of big data (Sahingoz *et al.*, 2019). Big data has enabled machine learning algorithms to discover more fine-grained patterns and to make more accurate and timely predictions than ever before (Zhou *et al.*, 2017). Deep learning techniques are used for object identification in images, the transcription of voice into text, matching news items and products with user interests and presenting relevant search results (Tyagi *et al.*, 2018). Deep learning architectures are composed of non-linear operations in multiple levels, such as neural networks (NNs) with hidden layers, or of complicated relational methods in reusable approaches (Montavon, Samek and Müller, 2018). The deep learning concept started with the study of artificial NNs (Vazhayil, Vinayakumar and Soman, 2018), and it has become an active research area in recent years.

The LSTM network has achieved excellent results in character recognition applications (Breuel *et al.*, 2013). It has also been used extensively in handwriting recognition, speech recognition and polyphonic music modelling, where the results have shown that its usage leads to an improvement in standard detection analysis when there is variance in the parameters (Greff *et al.*, 2017). It has also been used in language modelling to analyse speech in a speech recognition system, where it was found to show an improvement in confusion matrix over the recurrent NN (RNN) (Sundermeyer, Schlüter and Ney, 2012).

Li *et al*. (2019) proposed a model based on URL and HTML features to detect phishing web pages. They also designed a lightweight features HTML and URL, which they develop an HTML string-embedding features without using third-party services, which allows their model to work in a real-time detection application. They tested their method on a real-life dataset that consisted of over 100,000 URL and HTML features. The authors reported that their scheme was able to achieve 97.30% accuracy, a 4.46% true positive rate and a 1.61% false negative rate (Li *et al.*, 2019). Mishra and Gupta (2018) also focused on the text features of websites in order to propose a novel system for intelligent phishing detection to detect zero-day phishing attacks. For their model, the authors used the concept of uniform resources identifier and cascading style sheet matching. The authors reported that the proposed solution is very efficient in detecting phishing and zero-day attacks with a true positive rate of 93.27% (Mishra and Gupta, 2018). The above studies used the text features of websites to develop their phishing detection models. However, phishers can use other content on sites to evade detection.

Bahnsen *et al. (*2017) investigated the performance of LSTM in their work on a solution for phishing site prediction that uses URLs as input for machine learning models. The authors compared a feature engineering approach with the random forest (RF) classifier against a novel method based on an RNN. They used 14 features to build their lexical and statistical analysis of the URLs. They used an LSTM unit to build the model that receives as input a URL as a sequence of characters and predicts whether the URL is phishing or legitimate. They also constructed a dataset that consisted of two million phishing and legitimate URLs to train their model. They found that the LSTM model had an overall higher prediction accuracy compared to the RF classifier without the need for expert knowledge to create the features. Their approach was able to achieve an accuracy rate of 98.7% even without manual feature creation (Bahnsen *et al.*, 2017). However, again, their study only focused on the text features aspect of web pages. Hence, the performance of their model could be improved if other aspects such as the image and frame features of websites were included.

In recent years, the CNN has seen massive adoption in computer vision applications (Yu *et al.*, 2017). The CNN has also been used for feature extraction in the field of object recognition (Xu, Li and Deng, 2015). The CNN belongs to the family of multilayer NNs that have been developed for use with two-dimensional data, such as videos and images (Arel, Rose and Karnowski, 2010). The CNN is one of the most prominent deep learning methods where numerous layers are trained using a rigorous methodology. Recently, Yang, Zhao and Zeng (2019) proposed an approach for a multidimensional feature phishing detection solution that is based on a fast classification method using deep learning. In the initial stage, their solution extracts the features and character sequence of the URL and uses deep learning for quick classification; note that this step does not require third-party assistance or prior awareness of phishing websites. In the next stage, their solution combines the URL statistical features, web page text, code features and the quick deep learning classification into multidimensional features. The result of their experiment showed that their solution was able to achieve an accuracy of 98.99% and a false positive rate of just 0.5% (Yang, Zhao and Zeng, 2019). However, their scheme did not include the visual content of the websites in the classification model, even though phishers can easily modify images and thus prevent their website from being detected as a phishing website. Therefore, there is a need to include more features to improve the robustness of their scheme.

On the other hand, Vazhayil, Vinayakumar and Soman (2018) performed logistic regression using CNN, CNN-LSTM and a bigram to evaluate two datasets of URLs for phishing detection. They created a dataset from four different sources: MalwareDomainlist and MalwareDomain for malware URLs, and PhishTank and OpenPhish for phishing URLs. The dataset contained 60,000 URLs for the training phase and over 56,000 URLs for the testing phase. The dataset was used to train the CNN and CNN-LSTM models to detect phishing URLs. The LSTM algorithm was chosen because it can accept raw data URLs as input. The result of their experiment showed that the CNN-LSTM architecture was able to perform better than the other model, achieving an accuracy rate of about 98% for the classification of URLs (Vazhayil, Vinayakumar and Soman, 2018). However, the proposed technique only uses text-based features and could be improved if more features were added, and the parameters were optimised for more precision. The shortcomings observed in the above studies, therefore, informed the basis of our proposed IPDS.

In contrast, Yao, Ding and Li (2018) proposed a detection method with fast object recognition techniques using an improved R-CNN for small-scale identification. They decided to use a quicker R-CNN with a feature pyramid network for logo recognition because of the limited size of the two-dimensional code and because the size of the logos embedded into websites is also small. Their method is comprised of three processes: recognition and extraction, logo extraction, and recognition and identification. The authors reported that the result of their experiment showed that their proposed method was able to perform logo recognition more effectively than other methods (Yao, Ding and Li, 2018). However, their scheme only focused on images. If it had also used the text and frame features of the websites, the performance of their solution may have been improved. Similarly, Li, Wang and Kot (2017) proposed using the RNN and the CNN for image recapture detection in order to learn the deep representation of the images in order to extract discriminative and essential features of the intra-block and inter-block information of images (Li, Wang and Kot, 2017). On the other hand, (Xu, Li and Deng, 2015) used LSTM and CNN in combination to enable their solution to learn the temporal structure of videos in order to show how temporal features can be used for face anti-spoofing purposes and also to differentiate the genuine web page then attempt to identify fake websites.

Nevertheless, educating users remains a critical aspect of phishing detection because users need to be aware of phishing techniques and of how reputable organisations would communicate with them on the web and via email; the lack of phishing education among users is one of the contributory factors to phishing attack success (Jansson and von Solms, 2013). Due to the growth in cyberspace technology, computer users have a significant role to play in making the Internet a safer place for everyone because cyber attacks are targeted at achieving either financial or social gain (Arachchilage and Love, 2014) to the detriment of the user. On the other hand, some people undertake phishing activities for fun and a sense of accomplishment rather than for financial or social gain, but can also have adverse consequences for the user (Sharma, Meenakshi and Bhatia, 2017).

Phishing awareness has been improved through the development and use of online game training and email-based training to combat phishing attacks (Arachchilage and Love, 2013). However, there will always be some inexperienced users accessing Internet web browsers, and it is these users in particular who can quickly become phishing targets. Moreover, it is challenging to combat phishing solely through education because not only do users not read the educational materials, it is also hard to teach users how to make the right decision online. Therefore, continued user training and awareness may be the key to combating phishing attacks in organisations (Jansson and von Solms, 2013).

An organisation needs to secure its environment against phishing attacks and reduce its vulnerability, but it also needs to educate users on how to approach any suspicious email activities on their system and set up an automatic blocking mechanism to protect itself and its users from some known malicious sources. One of the tools that is used to achieve this aim is the phishing plugin.

Table I below presents a list of currently available phishing plugins together with the techniques they employ, their level of effectiveness and service type. Each of the plugins was developed for a specific browser, and not all are built for a cross-platform application. Thus there is an inherent weakness in the build of many plugins because end-users may have to use a browser that they are not used to when accessing content on the Internet and this reduces their efficacy.

Table I: Level of Effectiveness of Anti-Phishing Plugins

Plugin for Phishing	Algorithm	Browser	Effectiveness %	Service Type
GoldPhish	Google PageRank/OCR	IE	98	Free
Cloudmark	Matching	IE	94	Free
Microsoft SmartScreen	Matching	IE	95.9	Free
SpoofGuard	Image hash	IE	91	Free
Phishdentity	Google search-by-image API	IE	97.2	Research
PhisTackle	SHA1 hash	IE	91.3	Research
PhishGuard	HTTP digest authentication	Firefox	94	Research
PhishIdentifier	Jsoup library	Firefox	92	Research
PhishTester	Finite State Machine (FSM)	IE	97.1	Research
CANTINA+	TF-IDF	IE	98.06	Research
PhishAri	Random Forest	Chrome	92.52	Research
PhishShield	Jsoup library	Chrome	96.57	Research
PhishNet	Matching	Chrome	95.0	Research
PhishDef	Support Vector Machine (SVM)	Chrome	97	Research
Google safe browsing	Google PageRank	Chrome; Firefox	93.3	Free
PhishZoo	Fuzzy hashing	Chrome	96.10	Research
Seclayer	JQuery	IE; Chrome	91	Free
IPDS	LSTM+CNN	IE; Chrome; Firefox	93.28	Research

Table II shows the types of feature that the phishing detection model of each browser in Table I uses to identify phishing websites. As indicated in the table, the majority of the plugins use text and heuristic approaches in their scheme. The heuristic-based anti-phishing technique uses websites features such as text and frame content for phishing detection analysis to create a robust classification model (Lee and Park, 2016). Others use the blacklist/whitelist approach, which in simple terms, is similar to the use of signatures in antivirus solutions which maintains a blacklist of the sites that contain malicious content. However, blacklisting is reactive and can be evaded by the rapid recycling of blocked phishing web pages.

All the features and techniques listed in Table II were explored to develop the phishing detection and protection scheme proposed in this study. As these features have not been used together as a single solution, except in our previous work (Adebowale *et al.*, 2019), this combination approach is considered to be the key strength of our developed plugin.

Table II: Techniques and Features of Phishing Plugins

Phishing Plugin	Techniques/Features					
	AI	**Frame**	**Heuristic**	**Image**	**Text**	**Whitelist & Blacklist**
GoldPhish	No	No	Yes	Yes	Yes	No
Cloudmark	No	No	Yes	No	Yes	No
Microsoft SmartScreen	No	No	No	No	Yes	Yes
Netcraft (Customise)	No	No	No	No	No	Yes
SpoofGuard	No	No	Yes	No	Yes	No
Phishdentity	No	No	Yes	Yes	Yes	No
PhisTackle	No	No	No	Yes	Yes	No
PhishGuard	No	No	Yes	No	Yes	No
PhishIdentifier	No	No	Yes	No	Yes	No
PhishTester	No	No	Yes	No	Yes	No
CANTINA+	No	No	Yes	No	Yes	No
PhishAri	No	No	No	No	Yes	No
PhishShield	No	No	Yes	No	Yes	No
PhishNet	No	No	No	No	No	Yes
PhishDef	No	No	Yes	No	No	No
Google safe browsing	No	No	No	No	No	Yes
PhishZoo	No	No	Yes	No	Yes	No
Seclayer	No	No	Yes	No	No	No
IPDS	Yes	Yes	Yes	Yes	Yes	Yes

3 Methodology

A great deal of background knowledge and experience of phishing and an enormous amount of related information was gained during this study. The use of high-quality datasets in phishing detection classification plays a significant role in building phishing model classifiers (Zareapoor and Seeja, 2015). In this study, a variety of literature was reviewed, and a phishing experiment and a survey were conducted and analysed, all of which identified many techniques for phishing solutions. In addition, a quantitative method was used to obtain statistical results. As a result of these endeavours, we developed a proposed approach that uses a feature-based online and offline model. In our approach, a deep learning algorithm was used in a phishing website detection system that was based on LSTM and the CNN. In this study, the CNN and LSTM were combined to detect a variety of website elements in order to attempt to identify phishing websites more accurately. The LSTM algorithm was used to detect extracted features such as the text and frame content of web pages, while the CNN was used to analyse the image features of the websites(Xu, Li and Deng, 2015).

3.1 Long short-term memory (LSTM)

The LSTM algorithm was used to form part of the structure of the proposed scheme that takes the input from a URL as a character sequence and predicts whether the link is a phishing or legitimate website. The LSTM algorithm is an adaptive 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 cell. 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 (Bahnsen *et al.*, 2017).

Furthermore, LSTM units can 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 (Xu, Li and Deng, 2015). 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 helps in making the performance of LSTM directly comparable to other RNNs because training can be done by using standard backpropagation with time (Hakkani-Tür *et al.*, 2016).

3.2 Convolutional neural network (CNN)

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 (Babaee, Dinh and Rigoll, 2018). The CNN is superior to the NN in terms of time delay. Essentially, in the CNN, the weights are shared in a temporal dimension, which leads to a shorter computation time (Acharya *et al.*, 2018). The general matrix multiplication in the standard NN is therefore replaced in the CNN (Xu, Li and Deng, 2015). 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 (Babaee, Dinh and Rigoll, 2018). This type of model for the training of the architecture layers led to the success of the first deep learning algorithms (Arel, Rose and Karnowski, 2010).

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 performance (Yao, Ding and Li, 2018). Another benefit of the CNN model is the lower pre-processing requirement (Vazhayil, Vinayakumar and Soman, 2018). The use of the graphics processing unit has accelerated computing techniques and has been exploited to rapidly develop the computational requirements of CNN (Xu, Li and Deng, 2015). Hence, in recent times, CNN-based solutions have been applied to image classification, face detection, speech recognition, handwriting recognition, behavioural recognition and recommender systems (Acharya *et al.*, 2018).

3.3 Feature extraction

For the LSTM, features were extracted and stored as a dataset, which was used for training and testing. The phishing website data was collected from PhishTank and WHOIS between 2 November 2017 and 28

February 2018. The images that were extracted from legitimate and phishing websites were collected from 10 August 2018 to 30 December 2018 numbered well over 10,000.

In this study, the CNN and LSTM were 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 Figure 1. The aim behind this conceptualisation was to integrate the CNN, LSTM and a deep learning algorithm and apply them to the features extracted from websites to thus detect phishing activities more accurately. Based on a comparison between the features extracted and the knowledge model, the classification of legitimate and phishing sites can be achieved. Furthermore, websites can be 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 used in the classification process.

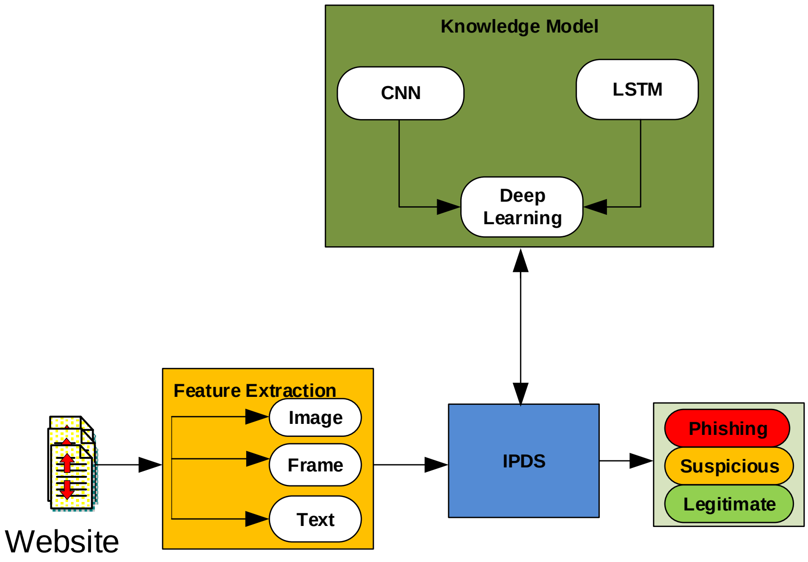


Figure 1: Structure of Intelligent Phishing Detection System (IPDS)

The overall conceptual framework for the offline IPDS structure using deep learning is presented in Figure 1. The concept involves using two deep learning algorithms, namely the LSTM and CNN, on different types of features that have been extracted from websites in order to better predict phishing activities. The extractor algorithm is used to extract the required feature from current websites. The knowledge model is used to compare the extracted features to determine whether the websites are phishing, suspicious or legitimate. The online approach consists of a user warning interface with three modules, and was built into the IPDS system. 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). If the features that was extracted from a website are loaded in the IPDS, it first check features with the knowledge model which is constructed with (LSTM+CNN) deep learning algorithm, the classification occurs. The first module in the online plugin warning system is the voice and text directive with a red colour status if the requested site is a phishing web page. The second is voice instruction and text direction with an amber colour status if the requested site is suspicious. The third is voice direction and text directive with a green colour status if the requested site is a legitimate page.

4 Experiment Setup

The section describes the experimental setup that was used for the feature selection and feature extraction methods. To train both the LSTM and CNN, a dataset was constructed that consisted of legitimate and phishing URLs. In total, a dataset containing 1 million URLs were used to train the LSTM algorithm. 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, more than 10,000 images (see sample in Figure 2) were collected from both legitimate and phishing sites. The image dataset was divided into two parts (70% and 30%, respectively) to train and test the CNN using holdout cross-validation.

Figure 2: Sample of images used to train the CNN

4.1 Data preprocessing

The raw data from both images and URLs contained a lot of background information and varied in length and size. Therefore, there was a need to pre-process this data to make it available for training the model. For the CNN architecture, the images were cropped from the sites based on the springing box and merely removed the wrong image. For the LSTM architecture, several websites features were collected and save in Microsoft Excel as comma-separated values.

4.2 Offline model development

The model was developed offline in MATLAB version 9.5 by using the deep learning toolbox. For the CNN architecture, there were three categories of layer: pre-training, fully connected and output layer. In this study, the AlexNet CNN was used. It is eight layers deep and can classify an image into over 1,000 object categories. The AlexNet CNN has a wide range of images as well as many learned rich features. The AlexNet network has an input image size of 227-by-227. In order to take full advantage of the capabilities of AlexNet, we retrained the pre-trained AlexNet network with the images that were obtained from various websites so that it would be able to classify new images. The AlexNet network was edited using the MATLAB version 9.5 deep learning toolbox. Later, the pre-trained learning is transfer and the fully connected layer output size was changed to the number of classes that needed to be categorised, i.e., three: legitimate, suspicious and phishing. Both the bias learning rate factor and the learning rate factor were set to 10. The first classification layer was deleted, and the new layer was connected. The newly connected classification layer was analysed, and the report showed zero errors. The new network was then exported into the deep network design. After that, the extracted image dataset was loaded into the image data storage and processed to extract the speeded-up robust features from all the images using the grid method to create a bag of features where the data was split into 70% for training and 30% for testing using holdout cross-validation. The images were resized to match the sizes of those in the pre-trained network input. In order to train the network, the exported edited AlexNet network layer from the toolbox was used to train the image collected and the options set. Then clustering was used to create a 1,000-word visual vocabulary (Figure 3). The model took 130 seconds to complete one epoch of the training procedure.

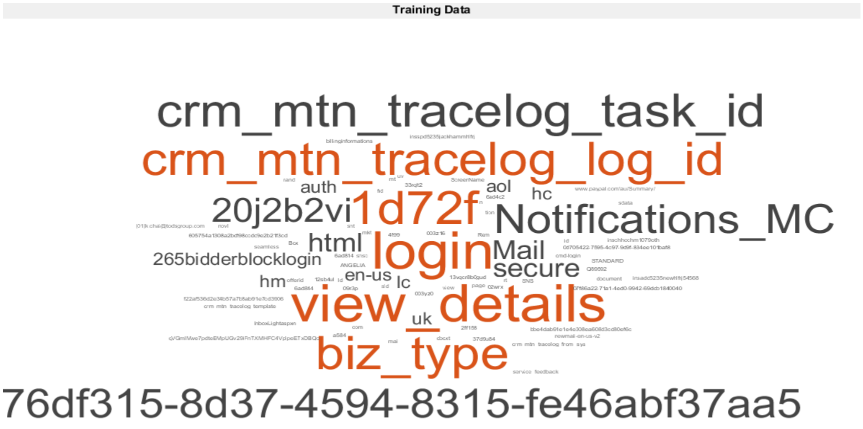
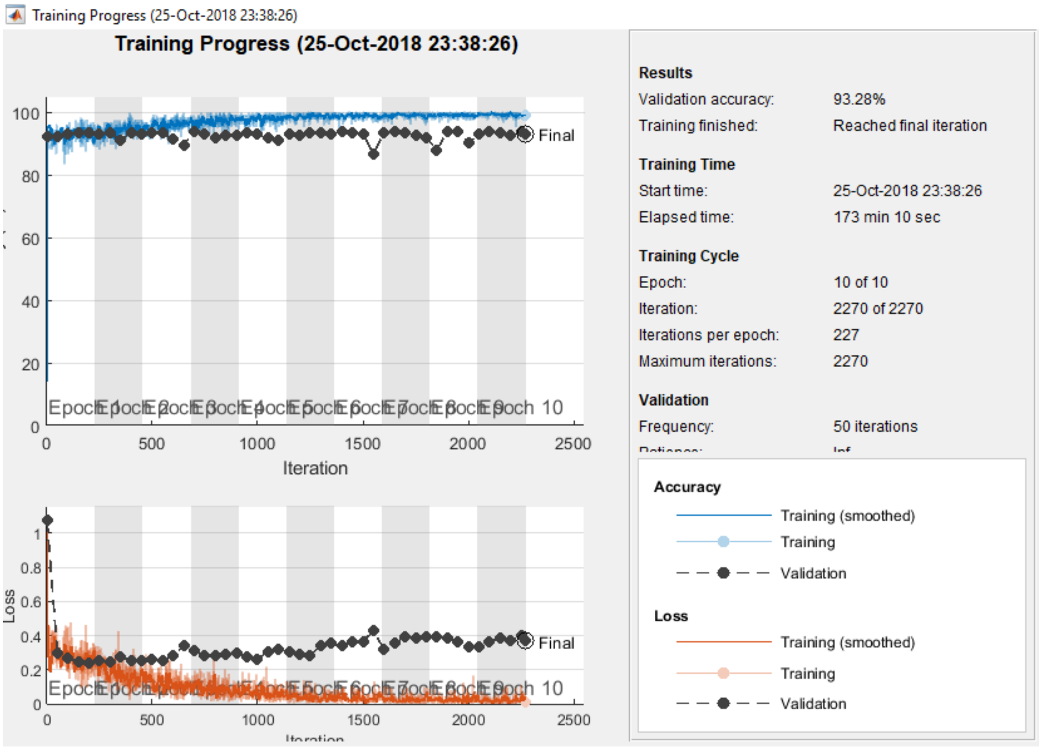


Figure 3: Visual vocabulary for the training dataset


Training Progress (25-Oct-2018 23:38:26)
Training Progress (25-Oct-2018 23:38:26)
100
80
60
40
20
0
Final
Epoch 1 Epoch 2 Epoch 3 Epoch 4 Epoch 5 Epoch 6 Epoch 7 Epoch 8 Epoch 9 Epoch 10
0
500
1000
1500
2000
2500
Iteration
Loss
1
0.8
0.6
0.4
0.2
0
Final
Iteration

Results
Validation accuracy: 93.28%
Training finished: Reached final iteration

Training Time
Start time: 25-Oct-2018 23:38:26
Elapsed time: 173 min 10 sec

Training Cycle
Epoch: 10 of 10
Iteration: 2270 of 2270
Iterations per epoch: 227
Maximum iterations: 2270

Validation
Frequency: 50 iterations
Patience: Inf

Accuracy
Training (smoothed)
Training
Validation

Loss
Training (smoothed)
Training
Validation

Figure 4: LSTM+CNN training and validation process

For the LSTM offline model, k-fold and holdout cross-validation were used. However, holdout cross-validation performed a little better than k-fold cross-validation (see Table IV). The dataset was divided, and holdout cross-validation was set to 0.7 for training and 0.3 for validation (Figure 4). Here, we use the testing dataset to calculate our evaluation metrics, and the rest of the data is used to train the model. This is the process of holdout cross-validation. The parameter was varied, but the 70% for training and 30% for testing gives the best result. The URLs were tokenised to separate each URL into a series of separate words, all of which were set into lowercase. The tokenised data was then encoded 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, the network architecture layer was tweaked that included using various parameter mentioned above to achieve better training accuracy.

5 Results and Analysis

The evaluation of the proposed method was performed based on traditional feature engineering, and on the classification algorithm methodology presented in section 3. Features based on the URLs and images features and websites elements were created. Then, the CNN and LSTM classifier were trained using one million URLs and over 10,000 images to build the proposed model.

Two series of experiments were performed for each evaluation method (LSTM, CNN and LSTM+CNN) to test their ability to identify legitimate, suspicious and phishing websites. In the first series of experiments, a time-based evaluation process was followed in which the time-stop time data was obtained for the point at which each method was able to classify all the legitimate, suspicious and phishing datasets. Then the training and validation process was repeated to determine the average time interval for each classification method.

In the second series of experiments, an accuracy-based assessment was performed in which all the legitimate, suspicious and phishing datasets were utilised to test the toolbar. In this assessment, the accuracy of each of the models was tested using the holdout cross-validation strategy. In the experiment, the overall classification accuracy result (Figure 5) for the proposed IPDS (LSTM+CNN) was 93.28% (Table III). The CNN achieved the best relative performance in terms of classification accuracy with a rate of 92.55% and that for testing was achieved by LSTM with 92.79% (Table III). Thus, the results showed that the accuracy of the proposed model (IPDS) was 93.28% with an F-measure of 93.29%.

In comparison, the model in our previous work (Adebowale *et al.*, 2019) was able to achieve an accuracy rate of 98.3%, and hence performs better in terms of accuracy than the IPDS method proposed in the current study. Nevertheless, the IPDS performed better in terms of the time taken to detect phishing websites, showing a 5-second improvement (25 sec vs 30 sec). Moreover, the IPDS improves on the performance of the model in the previous work due to the deep knowledge base that was built with the algorithm, which enables the IPDS to detect newly generated phishing websites without the need for manual feature engineering.

The results of the proposed scheme were compared with the approach proposed by Yang, Zhao and Zeng (2019), who used deep learning to develop a method that was able to produce a 98.99% accuracy rate in phishing detection. The proposed model was also compared with the method suggested by Bahnsen *et al. (*2017), which used LSTM for the detection of phishing sites and was able to achieve 98.7% accuracy for text-only feature detection. In our experiment, we considered this process, but fine-tuning of the features arranged together in the same attack pattern for training and testing purposes and assigning different weights with a reduction in some functions by removing the redundant elements used in their model. However, the accuracy rates reported in the above literature were not met by our study. However, the inclusion of more web page elements such as images and frames will further improve the robustness of the system proposed in the current study.

Table III. Comparison of Performance of Proposed Methods

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

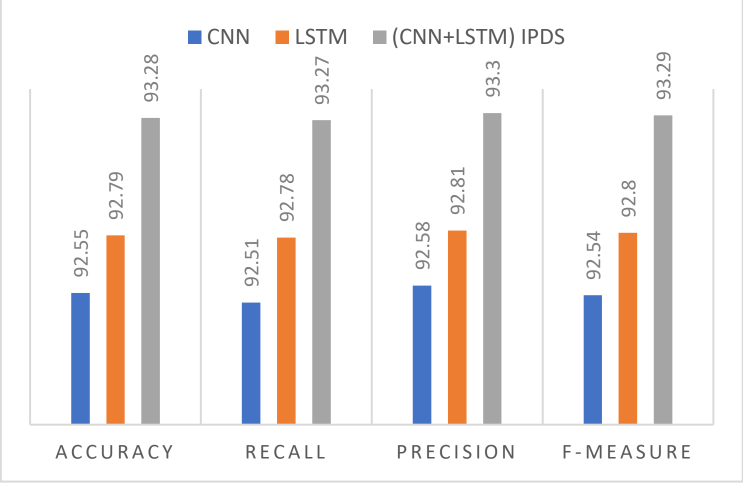


Figure 5: *Classification results for CNN, LSTM and IPDS (LSTM+CNN)*

Overall, 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 deep learning 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 IPDS model is able to report whether websites have any inherent threats based on its knowledge base of numerous web pages. If IPDS identifies that the web page poses a threat, the system informs the user about the possible presence of phishing content. Hence this scheme protects the online user from advanced malicious and phishing attacks based on a prior assessment of the site.

A primary aim of the experiment was to improve the detection accuracy and robustness of the system while at the same time reducing scrutiny time. Various methods in the literature could be used for training IPDS models, including 2-fold, 5-fold, 10-fold and holdout cross-validation. In this study, all the different folds of cross-validation were employed to train and test the accuracy and robustness of the proposed model. The result (Table IV) showed that the holdout cross-validation method produced the best result and was due to its effectiveness on existing datasets. According to our results, the classification approach is promising. The training and classification proved that it is possible to improve the classification process.

Table IV: Cross-validation Performance of Proposed Deep Learning Methods

Cross-validation	**Relative Performance**		
	LSTM (%)	**CNN (%)**	**(CNN+ LSTM) IPDS (%)**
2-fold	93.21	93.09	93.04
5-fold	93.20	93.25	93.06
10-fold	92.03	92.35	92.10
Holdout	92.79	92.55	93.28

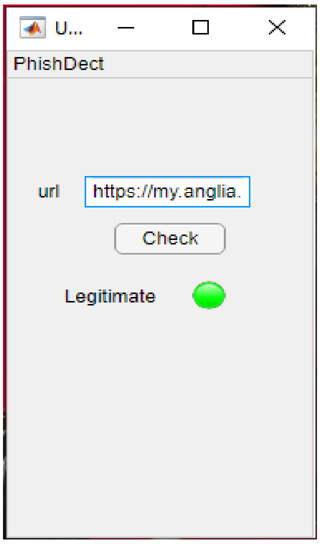
6 Evolution of Intelligent Phishing Detection System – Online Plugin

This section presents the development of the online plugin model of the IPDS. The feature-based online model has three essential parts: a feature extractor algorithm, the knowledge model and user warning interface. The features that were extracted in the experiments detailed in section 4 were used to develop a classifier. In addition, the LSTM+CNN algorithm was used in building a knowledge model that runs in the background as a toolbar comparing all the requested websites against the 35 features identified in the experiments in our previous work (Adebowale *et al*., 2019) to check whether the requested websites is legitimate, suspicious or phishing.

In the online IPDS model, the extractor algorithm is used to extract the required features from current websites. The knowledge model is used to compare the extracted features to determine whether the websites are phishing, suspicious or legitimate. The user warning interface has three modules: (i) voice generation with text directive with a red colour status if the requested site is a phishing web page, (ii) voice generation with text direction with an amber colour status if the requested site is suspicious and (iii) voice generation with text directive with green colour status if the requested site is legitimate. The plugin is implemented in the MATLAB version 9.5 AppDesigner toolbox. The online plugin was tested and evaluated on 1,000 phishing, 100 suspicious and 1,500 legitimate websites. We used the MATLAB version 9.5 AppDesigner toolbox to create a graphical user interface to evaluate the model. The checking process involved the user entering the URL link into the textbox. When the check button is pressed, the colour of the traffic light changes to correspond to the classification of the URL and the text also displays the classification value. Figure 6 displays the result for a legitimate site, Figure 7 shows the result for a suspicious site and Figure 8 illustrates the result for a phishing site.

As mentioned above, the plugin was evaluated on 1,500 legitimate URLs, 100 suspicious URLs and 1,000 phishing URLS. As stated above, the LSTM+CNN algorithm runs in the background as a knowledge module. The same procedure was used to test the toolbar performance on phishing and suspicious websites, where the algorithm checks whether the URL that has been requested is a legitimate website by comparing the newly typed URL in the text box against the stored features in the IPDS.

First, we tested the plugin on 1,500 legitimate sites. If no match is found when the site is checked against the knowledge base, then it is identified as a legitimate website, and the user warning interface displays a green colour status (Figure 6). At this point, it is safe for the user to continue in their task with peace of mind that the site to which they are submitting their confidential information is legitimate. In the experiment, this procedure was repeated 600 times with a validation dataset consisting of URLs so that most the URLs were tested to validate the performance of the toolbar and in each case, the result was observed and recorded (see Table V).



url: https://my.anglia.

Figure 6: Application interface for legitimate URL check

Figure 6 shows an example of a screenshot of one of the results produced by the toolbar for a legitimate site. As regards the time-based assessment of the toolbar’s ability to detect a legitimate website, the voice-generated user warning interface with a green colour status and a text showing the result was generated within 25 seconds before the page loaded.

Secondly, we evaluated the performance of the plugin on 100 suspicious URLs. As previously mentioned, the LSTM+CNN algorithm runs in the background as a knowledge module. The same procedure was followed in the testing of the toolbar on legitimate websites that described in the previous section, but in this test, the algorithm checks whether the URL requested is a suspicious website by relating the newly typed URL against the stored features in the IPDS. If a match is detected, and it looks like the URL is a suspicious website, the user warning interface included in the model shows an amber colour status and a text description is generated stating that the URL is "suspicious" (Figure 7) in order to alert the user to exercise caution.

In the experiment, this process was repeated 100 times, so are all 100 URLs were tested, and the performance was observed and recorded (Table V). An example of a screenshot of suspicious website result is shown in Figure 7.

The experiment to check for suspicious URLs was performed 8 hours per day over 2 days because there are only ever a few suspicious websites present online at any one time as they are short-lived, which makes this type of experiment a challenge to complete. As regards the time-based assessment of the toolbar's performance in identifying a suspicious website, the voice-generated user warning interface with an amber colour status and a text showing a warning were generated within 24.5 seconds to alert the user before the page loaded.

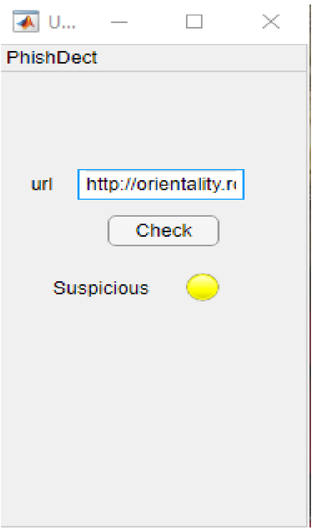
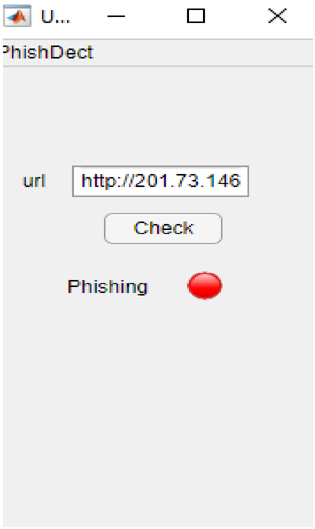


Figure 7: Application interface for suspicious URL check

Thirdly, we tested the plugin on 1,000 phishing websites. When a URL is typed into the address bar (Figure 8), the algorithm inspects whether the requested websites is a phishing link by comparing the current URL against the stored features in the deep learning classification algorithm. If a match is detected, and it is a phishing site, in order to alert the user a red colour status with a voice-operated user warning interface is activated and a text is generated showing that the status of the URL is "phishing". The performance of the toolbar in each case was observed and recorded, and in addition, screenshots were taken to validate the results. An example of a screenshot of a phishing website result is shown in Figure 8. This part of the experimental effort was carried out over 8 hours per day for 5 consecutive days. As regards the time-based assessment of the toolbar's ability to detect a phishing website, the voice-generated user warning interface with a red colour status and a text showing an alert were generated within 25.5 seconds to warn the user before the page loaded.



url http://201.73.146

Figure 8: Application interface for phishing URL check

6.1 Performance validation

One thousand phishing datasets were used to evaluate the performance of the plugin in terms of accuracy. Table V shows that the performance of the plugin in detecting phishing datasets resulted in a 93.8% true positive (TP) rate, a 6.2% true negative (TN) and an accuracy rate of 93.5%. When tested on 100 suspicious datasets, the toolbar achieved 94.5% accuracy, a 94.8% TP rate and a 5.2% TN rate. Also, when the plugin was tested on 1,500 legitimate websites, it achieved 91.8% accuracy, a TP rate of 92% and a TN rate of 8%. Overall, the toolbar was able to achieve an average accuracy of 93.28%. However, accuracy varied from a minimum of approximately 91% to a maximum of approximately 94.8%, which signifies that there was significant variation across the tested datasets.

Table V: Test Results for Phishing Website Detection by Toolbar Application

Status	No. of Websites	**Accuracy %**	**TP %**	**TN %**	**Average Result %**
Phishing websites	1000	93.5%	93.8%	6.2%	93.28%
Suspicious websites	100	94.5%	94.8%	5.2%	
Legitimate websites	1500	91.8%	92%	8%	

6.2 Time-based performance

In order to assess the time efficiency of the proposed scheme online, the toolbar was also validated against legitimate, suspicious and phishing websites. The experiment showed that the toolbar was able to produce a good result. The average time for websites to load content takes 60 seconds on a user computer system (Barraclough *et al.*, 2013). Based on this load time, we assessed the detection capability of the proposed plugin in three stages in order to determine the time the plugin needed to make real-time decisions (see Table VI).

Table VI shows that the details that are checked in each of the three stages. The first check by the plugin takes place within 10 seconds. During this initial timeframe it checks for the presence of the features that are most used in phishing websites. If any of these features are found on the web page, the user is alerted accordingly. Then the plugin goes to the second stage to verify the site against a specific list of features and it is expected to generate a result within 25 seconds. The third stage involves the checking of graphics. As graphics take more time to load, we decided that this check should be the last check performed by the plugin.

Table VI. Real-time Detection Stages of Proposed System

First-level check					**Time**
Using the IP Address	Long URL to Hide the Suspicious Part	Using URL Shortening Services	URL's having "@" Symbol	Redirecting using	10 s
Adding Prefix or Suffix Separated by (-) to the Domain	SubDomain and Multi SubDomains	HTTPS: HYPERText Transfer Protocol with Secure Sockets	Using Non-Standard Port	The Existence of "HTTPS" Token in the Domain Part of the URL	
Second-level check					
Domain Registration Length	Request URL	URL of Anchor	Submitting Information to Email	Abnormal URL	25 s
Age of Domain	DNS Record	Websites Traffic	PageRank	Google Index	
Number of Links Pointing to Page	Statistical-Reports Based Feature	IFrame Redirection	Server Form Handler (SFH)	Using Pop-up Window	
Tags	Disabling Right Click	Websites Forwarding	Layout Similarity	Style Similarity	
Third-level check					
Favicon	Image Size	Alternative Text	Mouse over	Login Form	15 s

The result of testing the plugin against these checking parameters showed that when a phishing website was requested, the plugin was able to alert the user within 24.5 seconds before the interface loaded its result. It alerted the user by using a voice-operated user warning interface with a text warning with a red colour status to denote the presence of a phishing website. In contrast, the plugin took an average running time of 25.4 seconds to identify a URL as a legitimate site before the web browser interface loaded its result. The plugin was able to identify a phishing URL more quickly than a legitimate URL because phishing URLs have some unique features that can be easily identified by the scheme.

Based on the above results of testing and validating the design concept on legitimate, suspicious and phishing websites, the proposed IPDS has a high level of accuracy and timely performance. To our knowledge, this study is the first to consider the LSTM+CNN algorithm for use in phishing detection and to apply it to a comprehensive set of features that includes image, text and frame content from all possible sources from a broad spectrum of websites.

The approach presented in this study is an extension of our previous work (Adebowale *et al.*, 2019) that used all the possible features of the image, frame and text content in terms of size and range. Hence our studies use a wider set of features compared to the majority of the previous studies, which used precise elements of websites such as blacklists and text features to develop anti-phishing toolbars (Bottazzi *et al.*, 2015). Indeed, as shown by Sharma, Meenakshi and Bhatia (2017), who surveyed 10 toolbars, the existing toolbars mostly use text features and blacklists.

7 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 IPDS. The deep learning algorithm was selected to develop the IPDS classifier for phishing website detection because of its capacity to perform deep analysis of both images and text. To evaluate the proposed hybrid approach, a dataset containing one million legitimate and phishing URLs collected from both the PhishTank and Common Crawl datasets were used. In addition, 13,000 features and 10,000 images were collected from both phishing and legitimate websites to build our offline model. The proposed IPDS gave excellent classification accuracy of 93.28%. The scheme has the ability to filter malicious websites based on the behavioural patterns obtained from previous data samples.

The IPDS was able to respond in real-time with great agility and could verify a URL in an average of 25 seconds before loading on the user's system. Also, our analysis revealed the advantages and disadvantages of both the CNN and LSTM methods. Overall, CNN performed better in terms of time, but on average, it was slightly less effective than LSTM. However, combining the two methods led to a better result in terms of accuracy with a shorter training time for the CNN architecture than for the LSTM model.

The primary contribution of this study is the integration of hybrid features that were extracted from text, images and frames and then used to develop a robust deep learning solution. This study is an extension work of our previous work that considered how best to integrate image, text and frame features with a deep learning algorithm (LSTM+CNN) to create a combined solution for a phishing detection scheme.

Future work will include improving the accuracy of the scheme and developing a web browser plugin based on a deep learning algorithm to detect web phishing across platforms and thus protect users in real-time.

References

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H. and Adeli, H. (2018) 'Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals', *Computers in Biology and Medicine*, 100, pp. 270-278.

Adebowale, M. A., Lwin, K. T., Sánchez, E. and Hossain, M. A. (2019) 'Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text', *Expert Systems with Applications*, 115, pp. 300-313.

Aggarwal, A., Rajadesingan, A. and Kumaraguru, P. 'PhishAri: Automatic real-time phishing detection on twitter', *eCrime Researchers Summit (eCrime)*, Las Croabas, Puerto Rico, 23-24 Oct. 2012: IEEE, 1-12.

Ahmed, A. A. and Abdullah, N. A. 'Real-time detection of phishing websites'. *7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, 13-15 Oct. 2016: IEEE, 1-6.

APWG (2019) *Unifying the Global Response to Cybercrime* [Online], Washington, D.C, USA: Anti-Phishing Working Group. Available at: https://docs.apwg.org/reports/apwg_ *identity theft* s_report_q2_2019.pdf.

Arachchilage, N. A. G. and Love, S. (2013) 'A game design framework for avoiding phishing attacks', *Computers in Human Behavior*, 29(3), pp. 706-714.

Arachchilage, N. A. G. and Love, S. (2014) 'Security awareness of computer users: A phishing threat avoidance perspective', *Computers in Human Behavior*, 38(2014), pp. 304-312.

Arachchilage, N. A. G., Love, S. and Beznosov, K. (2016) 'Phishing threat avoidance behaviour: An empirical investigation', *Computers in Human Behavior*, 60(2016), pp. 185-197.

Arel, I., Rose, D. C. and Karnowski, T. P. (2010) 'Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]', *IEEE Computational Intelligence Magazine*, 5(4), pp. 13-18.

Babaee, M., Dinh, D. T. and Rigoll, G. (2018) 'A deep convolutional neural network for video sequence background subtraction', *Pattern Recognition*, 76, pp. 635-649.

Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J. and González, F. A. 'Classifying phishing URLs using recurrent neural networks', *APWG Symposium on Electronic Crime Research (eCrime)*, Scottsdale, AZ, USA, 25-27 April 2017: IEEE, 1-8.

Barraclough, P. A., Hossain, M. A., Tahir, M. A., Sexton, G. and Aslam, N. (2013) 'Intelligent phishing detection and protection scheme for online transactions. (Report)', *Expert Systems With Applications*, 40(11), pp. 4697-4706.

Bottazzi, G., Casalicchio, E., Cingolani, D., Marturana, F. and Piu, M. 'MP-Shield: A Framework for Phishing Detection in Mobile Devices'. *International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, UK, 26-28 Oct. 2015: IEEE, 1977-1983.

Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A. and Shafait, F. 'High-performance OCR for printed English and Fraktur using LSTM networks'. *12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, 25-28 Aug. 2013: IEEE, 683-687.

Bullee, J.-W., Montoya, L., Junger, M. and Hartel, P. (2017) 'Spear phishing in organisations explained', *Information and Computer Security*, 25(5), pp. 593-613.

Chorghe, S. P. and Shekokar, N. 'A survey on anti-phishing techniques in mobile phones', *International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 26-27 Aug. 2016: IEEE, 1-5.

Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K. and Aparicio-Navarro, F. J. (2018) 'Detection of advanced persistent threat using machine-learning correlation analysis', *Future Generation Computer Systems*, 89(2018), pp. 349-359.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. and Schmidhuber, J. (2017) 'LSTM: A search space odyssey', *IEEE transactions on neural networks and learning systems*, 28(10), pp. 2222-2232.

Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y.-N., Gao, J., Deng, L. and Wang, Y.-Y. 'Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM'. *Interspeech*, San Francisco, USA, 8–12 September 2016, 715-719.

Hawanna, V. R., Kulkarni, V. Y. and Rane, R. A. 'A novel algorithm to detect phishing URLs', *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Pune, India, 9-10 Sept. 2016: IEEE, 548-552.

Jaison, F. and Francis, S. (2014) 'Phishing Website Detection: A Review', *International Journal of Computer Science and Mobile Computing, IJCSMC*, 3(2), pp. 696-699.

Jansson, K. and von Solms, R. (2013) 'Phishing for phishing awareness', *Behaviour & information technology,* 32(6), pp. 584-593.

Lee, J.-L. and Park, D.-h. (2016) 'Phishing Detection Using Web Site Heuristics', *International Information Institute (Tokyo),* 19(2), pp. 523-530.

Li, H., Wang, S. and Kot, A. C. (2017) 'Image recapture detection with convolutional and recurrent neural networks', *Electronic Imaging,* 2017(7), pp. 87-91.

Li, Y., Yang, Z., Chen, X., Yuan, H. and Liu, W. (2019) 'A stacking model using URL and HTML features for phishing webpage detection', *Future Generation Computer Systems,* 94(2019), pp. 27-39.

Mishra, A. and Gupta, B. (2018) 'Intelligent phishing detection system using similarity matching algorithms', *International Journal of Information and Communication Technology,* 12(1-2), pp. 51-73.

Montavon, G., Samek, W. and Müller, K.-R. (2018) 'Methods for interpreting and understanding deep neural networks', *Digital Signal Processing,* 73(2018), pp. 1-15.

Office for National Statistics (2017) *Crime in England and Wales: Year ending Dec. 2016*, London, United Kingdom: Office for National Statistics. Available at: https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/bulletins/crimeinenglandandwales/yearendingjune2016#whats-happening-to-trends-in-fraud (Accessed: 15 June 2017).

Sahingoz, O. K., Buber, E., Demir, O. and Diri, B. (2019) 'Machine learning based phishing detection from URLs', *Expert Systems with Applications,* 117(2019), pp. 345-357.

Shaikh, A. N., Shabut, A. M. and Hossain, M. A. 'A literature review on phishing crime, prevention review and investigation of gaps'. *10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA),* Chengdu, China, 15-17 Dec. 2016: IEEE, 9-15.

Sharma, H., Meenakshi, E. and Bhatia, S. K. 'A comparative analysis and awareness survey of phishing detection tools', *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT),* Bangalore, India, 19-20 May 2017: IEEE, 1437-1442.

Sundermeyer, M., Schlüter, R. and Ney, H. 'LSTM neural networks for language modeling'. *Thirteenth annual conference of the international speech communication association*, Portland, OR, USA, 9-13 September 2012: ISCA, 194-197.

Tyagi, I., Shad, J., Sharma, S., Gaur, S. and Kaur, G. 'A Novel Machine Learning Approach to Detect Phishing Websites', *5th International Conference on Signal Processing and Integrated Networks (SPIN),* Noida, India, 22-23 Feb. 2018: IEEE, 425-430.

Vazhayil, A., Vinayakumar, R. and Soman, K. 'Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks', *9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Bangalore, India, 10-12 July 2018: IEEE, 1-6.

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

Yang, P., Zhao, G. and Zeng, P. (2019) 'Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning', *IEEE Access,* 7, pp. 15196-15209.

Yao, W., Ding, Y. and Li, X. 'Deep Learning for Phishing Detection'. *Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom),* Melbourne, Australia, 11-13 Dec. 2018: IEEE, 645-650.

Yu, Y., Gong, Z., Zhong, P. and Shan, J. 'Unsupervised Representation Learning with Deep Convolutional Neural Network for Remote Sensing Images'. *International Conference on Image and Graphics,* Cham: Springer, 97-108.

Zareapoor, M. and Seeja, K. (2015) 'Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection', *International Journal of Information Engineering and Electronic Business,* 7(2), pp. 60-65.

Zhou, L., Pan, S., Wang, J. and Vasilakos, A. V. (2017) 'Machine learning on big data: Opportunities and challenges', *Neurocomputing,* 237(2017), pp. 350-361.

Zuhair, H. and Selamat, A. 'Phishing classification models: Issues and perspectives', *Conference on Open Systems (ICOS),* Miri, Malaysia, 13-14 Nov. 2017: IEEE, 26-31.