National College of Ireland

Phishing Detection Using Convolutional Neural Network and ADADELTA

MSc Internship
CyberSecurity

Tejas Umakant Phade
Student ID: 18195709

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Tejas Umakant Phade
Student ID:	18195709
Programme:	MSc CyberSecurity
Year:	2019-2020
Module:	MSc Internship
Supervisor:	Niall Heffernan
Submission Due Date:	17-08-2020
Project Title:	Phishing Detection Using Convolutional Neural Network and ADADELTA
Word Count:	6371
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	Tejas Umakant Phade
Date:	14th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Phishing Detection Using Convolutional Neural Network and ADADELTA

Tejas Umakant Phade
18195709

Abstract

The internet has been integral and indispensable part of people's life to do mundane tasks or to communicate personal/sensitive information or to use educational, medical, financial services. Since there is an influx of inexperienced users, the internet brings serious security problems and these vulnerabilities are exploited by attackers. Novice users are naïve and phishers use phoney web pages to lure and deceive them resulting into gaining their information which has costed users a lot of their fortune; this technique is known as Phishing. Decades have been devoted in developing novel technique to detect phishing website. Even though superior performance can be achieved through state-of-the-art solutions, it demands substantial amount of manual engineering and does not provide guaranteed results. In this study, we focus on design and development of phishing detection solution based on deep learning, leveraging Convolutional Neural Network (CNN) and ADADELTA algorithm. The developed solution is a pure image-based approach with addition of similarity detection has been taken to avoid non-text phishing tricks such as HTML Contents or Flash objects. Accuracy of 96% is shown with the proposed model.

1 Introduction

Being from the family of social engineering attacks, phishing is used to steal victim's personal information ranging from credentials to card details with the help of spoofed Web Pages which has been going around since early 90's [1]. Not only large-scale organizations but also small-scale organizations and individuals are vulnerable to phishing attacks because of which the expenditure on Cyber Security was $ 124 Billion in 2019 and as per some research it might exceed $ 1 Trillion in 2021 [2] [3]. Legitimate websites are mirrored by phisher with the help of tactics and approaches as they have wide array of tools and because novice users are unaware and do not poses sound knowledge about phishing, they often get caught in the phishers trap resulting in losing their valuable information. Approximately in every 20 seconds a new phishing websites is created and according to APWG, nearly 74% of phishing websites are hosted with HTTPS/SSL certificates which makes it harder to distinguish between phoney and authentic website [4] [5]. Nearly 470,000 complaints have been made in the last year to the Internet Crime Complaint Centre by phishing victims and recorded more than $ 3.5 Billion losses to businesses and individuals hence, detecting these phoney Web Pages are crucial [6].

Detection of phishing attacks is generally classified into two approaches: the user training approach and software classification approach [7] [8]. As per the name, in the

first approach, the user is trained to identify difference between phishing Web Page and legitimate one [9] [10]. User training approaches are aimed towards training the users for finding intricate details in legitimate and phishing websites but it can be troublesome as not all users come from Information Technology background. In case of software classification, this approach is widely used to identify phishing websites with the help of different techniques such as blacklisting URL's (Uniform Resource Locator) or the heuristic or visual similarity approach [11] [12] [13]. Blacklisting URL's which are targeted towards phishing websites helps in stopping users from falling for traps set by phishers, but with the ease of hosting a website there are multiple phishing website hosted per minute makes it hard for users to keep their blacklist updated. In case of heuristic approach, the false positive rates were high causing it to misclassify the legitimate Web Pages and proving itself to be highly unreliable. Anthony Y. Fu et al. (2006) and Kuan-Ta Chen et al. (2009) showed phishing detection techniques with the help of global and local features respectively [14] [15]. Since traditional detection methods have more margin of error and ineffective to detect modern phishing websites, visual similarity yields better results.

Analysing and identifying images can be done in various other ways such as Deep Learning (DL). This concept started with the study of Artificial Neural Networks (ANNs) which has become active research field in past few years. There is a requirement of neurons for building a standard Neural Network (NN) and given adequate weights, ANNs behaves as needed. Even though training the ANN with backpropagation makes it useful and accurate, the testing data might be unsatisfactory giving rise to new technologies. Convolutional Neural Network comes under DL algorithm which takes input of an image and performs tasks such as assigning weights or biases to different objects or aspects in the image whilst differentiating one from other. There is little to no pre-processing requirement in CNN and it has the ability to learn characteristics and filters. Being inspired by Visual Cortex, the CNN has connectivity pattern of Neurons in Human Brain and it is able to successfully capture the Temporal and Spatial dependencies. The CNN can also be used to fulfil the role in reducing image form making it easier to process without any loss of image features. The CNN can have multiple layers, first being a convolutional layer, which has the responsibility to capture any low-level features of an image such as gradient orientation, edges or colours and it can be adapted to capture high-level features while providing understanding of image. Secondly, pooling layer has the responsibility to reduce spatial size to decrease requirement of computational power. It can also be used to extract dominant features to maintain the training efficiency of the model.

In this report, we're seeking the answer to *Can Convolutional Neural Network and ADADELTA be used to detect phishing Web Pages?*

This research proposes a phishing detection artefact based on visual similarity. The paper is organized as follows, Section 1: Introduction is aimed towards the understanding and motivation behind phishing and phishing detection technologies. In Section 2: Related Work, we'll discuss findings of previous researches on the topic of phishing detection and visual similarity and it's relation to Convolutional Neural Network. In Section 3: Research Methodology, we will look at the model followed to develop the artefact and leveraging the literature we'll justify the choices. In Section 4: Design Specification, the underlying architecture for our developed solution is discussed. In Section 5: Implementation, the proposed model's implementation along with algorithm is shown. In Section 6, we'll discuss the Evaluation and the analysis of Experimental Results followed by Discussion and Limitations. Lastly, in Section 7 we will look at the Conclusion and

shed some light on possible Future Works.

2 Related Work

Internet is ubiquitous and it is used in completing minuscule to colossal tasks such as browsing for information to transferring of funds. These tasks are mainly carried out on a browser making it easier for the user to interact. Because browsers are customizable, anyone can create their own program for it known as "Extension" which gives browser the ability to perform different tasks. Novice users are unaware making them suitable to fall for phishing attacks giving rise to phishing detection extensions.

Thomas Raffetseder et al. (2007) in their paper *Building Anti-Phishing Browser Plug-Ins: An Experience Report* did a comprehensive research and comparison on different browsers and their abilities to help in mitigation of a phishing attack by keeping users' sensitive information secure [16]. At the time of their research, *Google-Chrome* was not released and the concept of plug-ins/extension was new. Even though they faced difficulties while implementing it, their work was used as foundation for further research.

In research paper *Anti-Phishing in Offense and Defense* authors Chuan Yue et al. (2008) created a client-side anti-phishing tool which stored the victim's real credentials while feeding the login system bogus ones and in-turn gives opportunity to the legitimate website in detecting phishing Web Page [17]. *PhishGuard: A browser plug-in for protection from phishing* is a study from author Yogesh Joshi et al. (2009) where they proposed a novel algorithm in which random credentials are submitted to a login page and the response is analysed to identify a phishing Web Page [18]. In the proposed architecture, authors took advantage of HTTP (Hyper Text Transfer Protocol) Digest Authentication Scheme in which an HTTP Post or Get request has been sent to server to gain it's status and if active then random credentials will be sent and the response from server will be analysed to determine the authenticity of Web Page. In research paper *Which web browser work best for detecting phishing* the authors Noman Mazher et al. (2014) ran developed solution thorough tests on different set of browsers and concluded *Google-Chrome* to be the better option to select as a safe browser [19]. Browser extensions can be used in various ways for betterment of user but malevolent person can use it to fetch user's information. Gaurav Varshney et al. (2018) while researching for *Browshing a new way of phishing using a malicious browser extension* created an extension with nefarious purpose and concluded their hypothesis by fetching users HTML data and keylogging information [20]. *Malicious browser extensions: A growing threat: A case study on Google Chrome: Ongoing work in progress* was another paper in which Gaurav Varshney et al. (2018) using their credibility provided some solutions and suggestions providing help in thwarting the attack possibility [21].

The motive of phisher is to emulate real Web Page by any means which leads them in using advanced techniques to avoid being detected from various detecting mechanisms. Yeu Zhang et al. (2007) in their research paper *CANTINA: A Content-Based Approach to Detecting Phishing Web Sites* leveraged Robust Hyperlinks concept to use in detection of phishing giving them an edge over other solutions [22]. Guang Xiang et al. (2007) did their further research into this field creating *CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites* by analysing content majorly focusing on text [23]. However, phishers constructed Web Pages purely from an image proving the content-based detection infeasible. These sophisticated phishing

attacks gave an emergence to visual similarity based phishing detection.

While researching the topic, Neil Chou et al. (2004) came up with "SpoofGuard" in their paper *Client-side defence against web-based identity theft* found a pattern in phishing Web Pages of copied images including logos, buttons or banners and implemented image check of suspected Web Page to cross verify the source of those images and if higher the number of suspected images, higher the probability of page being phishing [24]. Authors raised a concern in which a slight modification to an image can yield false positive results and demanded a standard image should be followed to be recognized by optical character recognition (OCR). The scarcity of optimal image hashing solution deemed SpoofGuard ineffective for advanced phishers techniques.

Since the region of visual similarity was still novel and unexplored many researchers started to work on optimal solution to mitigate phishing Web Pages. Liu Wenyin et al. (2005) provided a novel solution for phishing detection by the means of decomposing Web Pages into salient blocks and the similarity is measured into three levels; block, layout and overall style similarity [25]. Phishing page is evaluated by comparing in similarity threshold, if any mismatch occurs then particular Web Page is flagged as phishing. While working on detection of phishing Liu Wenyin et al. (2006) leveraged segmentation algorithm based on Document Object Model (DOM) representation in which suspected Web Page is again segmented into blocks but verified on a granular level and the author classifies them as: block level, layout level and overall [26] [27]. The block level similarity is the average of all visual similarities, layout similarity is achieved by calculating the ratio of weighted number of matched blocks to the total blocks in legitimate Web Page and overall style similarity is defined by format of the Web Page such as font family, line spacing. Even though the solution is advanced, if method is faced with two Web Pages having similar appearances but with different DOM representation, it will be proven infeasible.

Researchers were finding various means to stop phishers from damaging victims and doing so, Yingjie Fu et al. (2005) presented an anti-phishing approach based on global visual similarity, in their paper *EMD based Visual Similarity for Detection of Phishing Web Pages* [28]. The architecture of solution was such that suspected and target Web Page are pre-processed into low resolution images and a signature consisting features such as dominant degraded colours, coordinate centres and weights of page i.e. corresponding degraded histogram of colours are extracted. Authors introduced a concept of Earth Mover's Distance (EMD) algorithm by which a similarity value can be obtained and if the value is beyond the predefined threshold, the suspected Web Page can be flagged as phishing Web Page. Another set of research has been done on EMD by Anthony Y. Fu et al. (2006) in their paper *Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)* wherein with similar studies but on a higher scale in which they used more than 10,000 suspected Web Pages to prove their hypothesis [14]. This technique tackled the obfuscation scam of phishing Web Page. However, the sheer number of target legitimate Web Page and suspected Web Pages are huge, increasing the probability of false positive cases and making the method highly unreliable.

While searching for the perfect and novel solution, Eric Medvet et al. (2008) presented a paper *Visual-Similarity-Based Phishing Detection* in which text pieces and their styles, embedded images and overall appearance of Web Page; these three features are extracted from Web Page [29]. The first feature of Web Page includes textual content, colour, font size, font family and their relative position coordinate on page. Secondly,

embedded images are extracted such as image, their source address, relative position coordinates on the page and even the 2D Haar wavelet transformation. Finally, rendered image corresponding to the visible region including colour histogram and 2D Haar wavelet transformation are extracted. They are then compared on the basis of feature vectors and if the values of two vectors are similar, then the page is considered as phishing. Even though it uses advanced techniques to mitigate phishing, it still struggles to deal with phishing pages including copy of images from legitimate Web Page.

Another means of phishing detection by comparing local similarity was published by authors Kuan-Ta Chen et al. (2009) in their paper *Fighting Phishing with Discriminative Keypoint Features* where they discuss taking a snapshot of suspected Web Page and target legitimate Web Page after which extraction of Harris-Laplacian corners is achieved which are known as keypoints of the images [15]. Each keypoint is assigned a descriptor, computed from Lightweight Contrast Context Histogram (L-CCH). Based on their coordinates, clustering of keypoints at each image using "k-means" algorithm is achieved. In accordance of these cluster, sets of two L-CCH descriptors are respectively matched yielding us a similarity value. If the similarity value is beyond the predefined value, suspected Web Page is classified as phishing Web Page. Focusing local similarity, authors do not consider global similarity and since "k-means" value is calculated by number of clusters, we cannot predict an upcoming unknown Web Pages cluster value since they are different for each Web Page. For complex phishing Web Page, distribution of keypoints are irregular making it impossible to gain keypoints for "k-means" cluster computation.

Detection of phishing Web Pages on the basis of local similarity is further researched by Guang Xiang (2013) in their research paper *Toward a Phish Free World: A Feature-type-aware Cascaded Learning Framework for Phish Detection* [30]. Author relied on checking the inconsistency between the identity of contained logo claims and real identity that domain claims. First and foremost, the neural network is trained using legitimate logo images and classified on five features: height, width, position, keywords related to logo and finally image tags concurrent text words and title of the page. Each image is extracted and the probability of it being a logo is computed and if it exceeds 0.5 then the image is recognized as candidate logo. Later, a duplicate image matching technique is used on the candidate logo image to compare it with set of protected official logos and if the domains are mismatched, then the Web Page is flagged as phishing Web Page. As this is also an image-based approach, it can misjudge the Web Page yielding false positive results.

Currently, there are substantial amount of ways in identifying and verifying images using technology. It can be a hybrid model based on mixtures of multiscale deformable part models proposed by Pedro F. Felzenszwalb et al. [31] which has been widely used in other researches. Similarly, there are multiple image classification algorithms, such as Vladimir Vapnik et al. (1995) introduced Support Vector Networks [32], "OverFeat" which was proposed by Pierre Sermanet et al. (2013) using deep learning approach for localization achieving state-of-the-art detection task. Also, some people prefer the use of Random Forest (RF) [33] introduced by Leo Breiman (2001) and some work with Radial Basis Function Network by Mark J. L. Orr [34]. However, recently the use of Convolutional Neural Network has been used for analysing and processing image data. The technology of deep learning attracts researchers since it can be used to perform face and object recognition along with other type of image recognition such as handwritten characters with higher precision than the traditional learning systems.

A mechanism for visual pattern recognition i.e. "Neocognitron" is proposed by Kuni-

hiko Fukushima (1980) in his paper *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position* where geometrical similarities are measured [35]. Author introduced number of cascade connection in modular structure wherein each layer consisted of similarity detection cell S-cells and a complexity detection cell C-cell. The model has ability to self-train or unsupervised learning. The work and research by author paved the way for future development. In later research Kunihiko Fukushima et al. (1982) took their research further in *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition* and concluded that deepest layer of network is not affected by the position shift of objects but intermediate layers close to input layers are affected [36].

Taking the leverage of Back Propagation algorithm by David E. Rumelhart et al. (1987) in their chapter *Learning Internal Representations by Error Propagation*, introduced the ability and usefulness of the algorithm [37]. In the era of handwritten image recognition applications Yann LeCun played a vital role. Along with other authors, Yann LeCun et al. (1989) proposed and applied an approach to recognize handwritten zip codes in the U.S. Postal Service while only utilizing a single network [38]. In the process of image recognition, Yann LeCun et al. (1998) proposed a novel solution using convolutional neural network and gradient-based method minimizing performance measure of rendering handwritten characters on cheques [39]. The author added global training techniques for recognizing cheques and it has been deployed commercially to read millions of cheques per day. Li Chen et al. (2014) along with authors argued that CNN should be used for selecting local receptive fields and got 96% accuracy in recognition of handwritten Chinese characters [40].

In a contest of developing efficient model for image recognition, Alex Krizhevsky et al. (2017) introduced *ImageNet classification with deep convolutional neural networks* which included 60 million parameters and 650,000 neurons along with three of the fully connected layers and utilizing GPU efficiently [41]. Authors developed a new method called "Dropout" making them win the contest and getting error rate of 15.3% while runner up had 26.2% proving their solution substantially efficient. Florian Schroff et al. (2015) developed and proposed a novel triplet mining method, implementing the face verification and recognition technique known as *FaceNet* which uses asfeature vectors [42]. Authors used deep convolutional network to directly optimize embedding achieving 99.63% accuracy.

CNN has made its impression amongst other things including hand posture and gestures recognition. By using pre-filtered images of Gabor Filter and using manually trained dataset of 6000 labelled images, Dennis Núñez Fernández et al. (2017) proposed *Hand Posture Recognition Using Convolutional Neural Network* with using single camera [43]. Author presented an idea of identifying hand posture on the basis of wrist position. As the scope of implementing CNN into healthcare benefits, Norah Alnaim et al. (2019) presented an artefact aiming for stroke survivors since their speaking ability can be impaired and to overcome it, authors presented *Hand Gesture Recognition Using Convolutional Neural Network for People Who Have Experienced A Stroke* with accuracy of 99% in understanding the gestures [44].

As the time progressed, CNN is being used in multiple authentication phases as well. Zhengbing Hu et al. (2018) used the capability of convolutional neural network in biometric scan on the basis of face geometry along with optimized method for face recognition and their results are proved experimentally [45].

To achieve the required results, Matthew D. Zeiler et al. (2012) in their paper

ADADELTA: AN ADAPTIVE LEARNING RATE METHOD made various modifications to gradient descent algorithm [46]. Also, authors overcame issues present in ADAGRAD such as sensitivity to the hyperparameter and to avoid continual decay of learning rate [47] . Authors while designing Adadelta, were highly inspired by Yann Lecun et al. (1988) work in Back-Propagation algorithm and leveraged this property as well [48]. ADADELTA is proven to have robust learning rate method and can be applied to variety of solutions.

3 Research Methodology

The Research Methodology is known as the philosophical framework within which the research is conducted [49]. Generally, research methodology is conducted to formulate an idea of how the whole research is carried out. In this section we will discuss the methodology of artefact developed to detect phishing website. To accomplish the desired outcome, we've followed Secure Spiral Model introduced by Daljit Kaur et al. (2012) which introduces aspect of risk analysis and focuses on security in every phase of its development cycle [50]. It promotes development of software in quick and small steps which are based on continuous iteration allowing organizations to release updates more frequently. We've considered Agile and Waterfall Model also, but since they do not focus on aspect of risk analysis or security, the Secure Spiral Model was selected. The secure spiral model enforces to apply security in early and each stages of development.

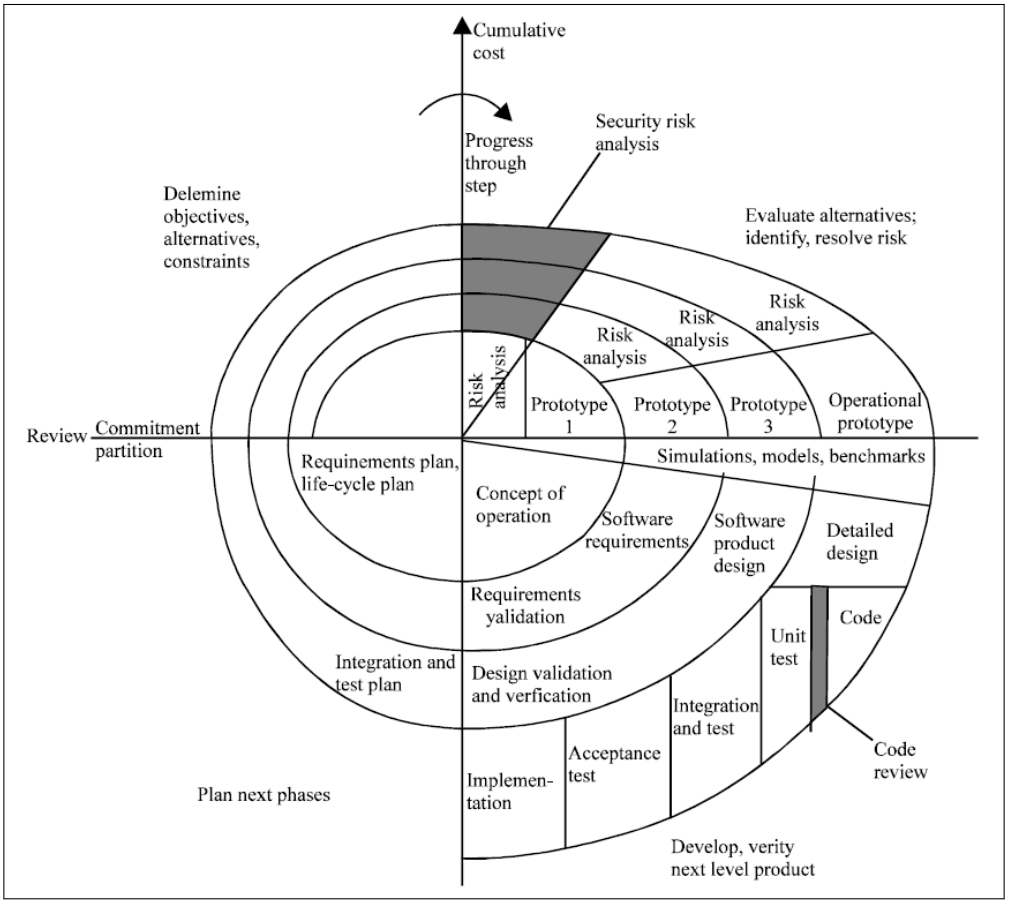

Figure 1: Secure Spiral Model [50]

3.1 Phase 1: Requirement

The proposed artefact is purely based on visual similarity detection of legitimate Web Pages. After reviewing previous researches we've decided to use Convolutional Neural Network because of its usability and versatility in image processing. The CNN has an architecture which is discriminative and it shows satisfactory performance in terms of two-dimensional data such as videos or images. Having a superior concept than Neural Network (NN), CNN weights are shared in temporal dimension leading to decrease in computation time. Since the general matrix is replaced in CNN compared to NN, the complexity decreases thereby reducing weight of the network. Furthermore, using back-propagation algorithm increases performance of network because of decrease in number of parameters present in CNN topology. Because the CNN leverages machine's Graphics Processing Unit (GPU), there is an increase in computation and it has lowered pre-processing requirement.

To train the extension, we've selected ADADELTA, it is introduced for tuning the learning rate dynamically in the CNN training process whilst ADADELTA can speed-up the convergence rate achieving good training results. Even though the idea of ADADELTA is derived from ADAGRAD, the ADADELTA is more robust and can be applied to any complex solution [47] [51].

In the plethora of coding languages, JavaScript has been selected to code our artefact extension. In the phase of Requirement, we've tried multiple coding languages, but JavaScript gives the freedom and ability to perform certain required task. In the proposed artefact, detection of phishing website is accomplished by image processing and collection of data is a vital process to get the desired results. The experiment has been conducted on selective 30 websites from top 100 visited websites globally [52]. The sample images have been gathered manually by statically adding domain into extension code and capturing the image followed by storing them in respective folders; this makes it a Primary Data collection of image data and the capturing of images was done manually to minimize the chance of an error. These sample images will be then learned by the neural network to identify and notify user about the presence of phishing Web Page.

The vital part of any extension is to work in browser and to use Deep Learning into browser along with CNN, ConvNetJs is used [53]. The ConvNetJs works independently and is not reliant on software, compilers or GPUs (Graphic Processing Unit). In the training of dataset, ConvNetJs uses multiple layers in order to train our extension such as Convolutional Layer, Pooling Layer. Training of CNN along with ADADELTA and ConvNetJs is a time-consuming task which depends on dataset and configuration of machine it is being trained on. Followed by training the dataset, the machine can then be used to detect the phishing page.

3.2 Phase 2: Design

In the below figure we've shown a holistic approach of our artefact. Even though the proposed system does not have any role-based implementation but the idea of "Admin" and "User" stems from the usability perspective as user will not have any control over the code of the artefact.

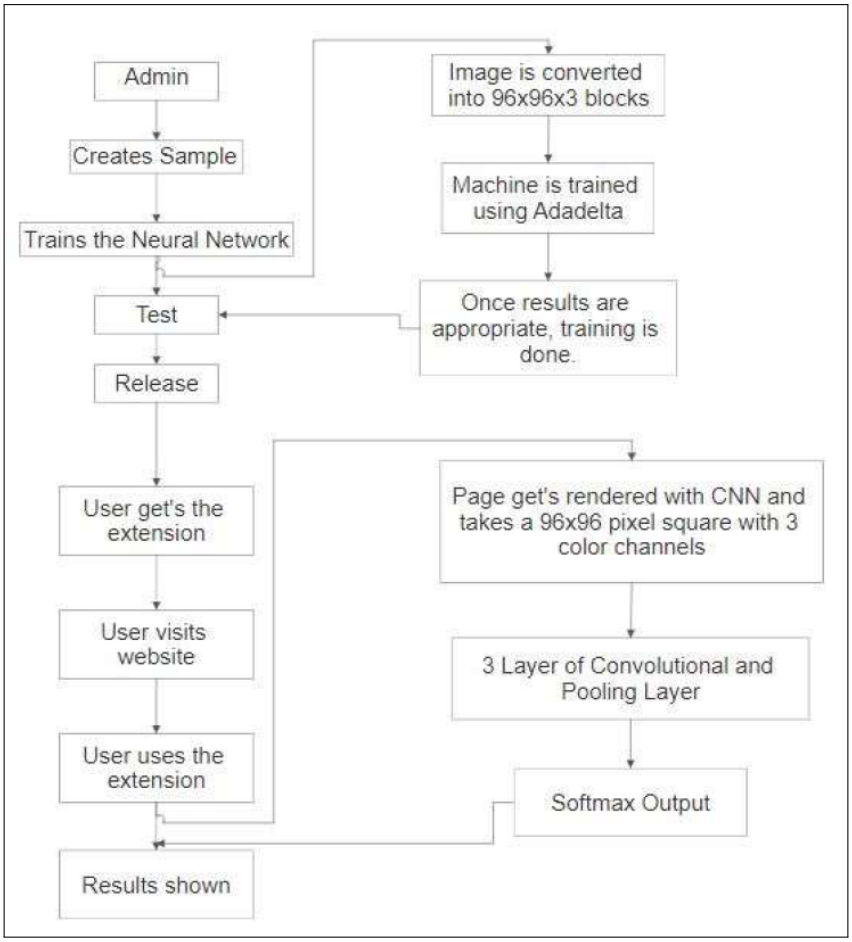

Figure 2: Holistic View of Artefact

3.3 Phase 3: Coding

As we're following the Secure Spiral Model, basis of this model is to follow security from early stages of an artefact development. Since we're using JavaScript in developing our extension, secure coding standards by OWASP and SANS are followed to minimize any risks [54] [55]. Depending on the usage, the files are coded mainly in JavaScript or HTML (Hyper Text Markup Language) or CSS (Cascading Style Sheets) since some usability of other languages can be leveraged to our advantage. To train the machine, we've used ConvNetJs complementing our choice of coding language. To optimize the usability of our developed solution, in some cases the Bash Script has also been used. We've utilized one input layer, three alternate convolutional and pooling layers and lastly one softmax layer to compensate for regression which is also known as loss layer and ReLU is used as an activation layer.

3.4 Phase 4: Testing

Once we've trained the machine, our extension is ready to be tested. We've used selective 30 websites from top-100 websites as our dataset and the tests are based on them [52]. Manual testing is required to minimize any error in identifying flaws and noting them is necessary since testing and keeping record is vital for further changes or upgrades in our

extension. By going through our excel sheet, we've recorded any flaws and rectified them before reaching final deployment phase.

3.5 Deployment & Review

In the phase of deployment, risks are analysed as we're following Secure Spiral Model and later artefact is deployed for public use. Since we've tested our artefact manually, some issues can arise in later use and having reviews of users can be useful for further development of our product. Because we follow Secure Spiral Model, we can take users review as input into next requirement phase and start the cycle yielding in better version of our artefact with new modification fulfilling their requirements.

4 Design Specification

In this section of paper, we will look at the design of our proposed system and discuss in detail the intricate details included in our artefact. Being an extension, this artefact does not have roles but we've divided the system into two pseudo parts, 'Admin' and 'User'. These two roles differ in the tasks they perform.

4.1 Admin

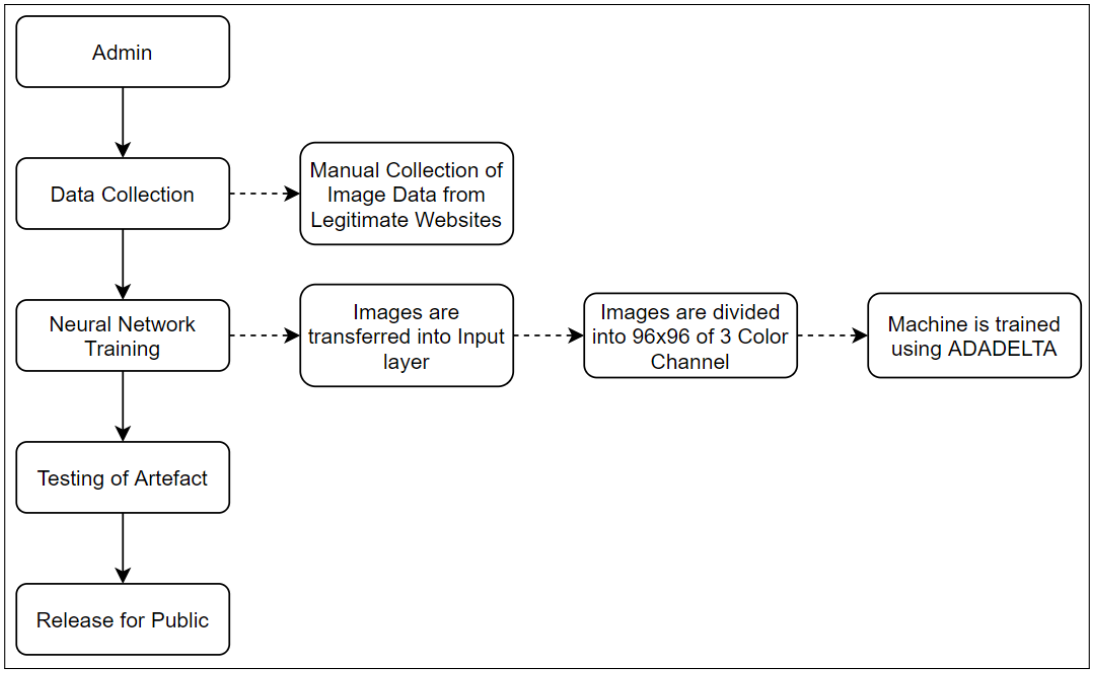

Figure 3: Admin Workflow

The developer or moderator of this artefact has several tasks to fulfil. Admin statically and manually collects data (Images) and stores it in the respective folder of samples for website. After ample amount of data is collected, the Neural Network is trained to reach desired accuracy. The images are scaled down and login form is cropped which is then converted into 96x96 square of 3 colour channels. Neural Network is then trained

with the help of these images and using ADADELTA algorithm. After the network is fairly competent at identifying the configured labels, the Neural Network trainer starts to increase the ratio of negative samples giving the network more resilience. We've also added a roulette selection process while feeding positive case samples.

4.2 User

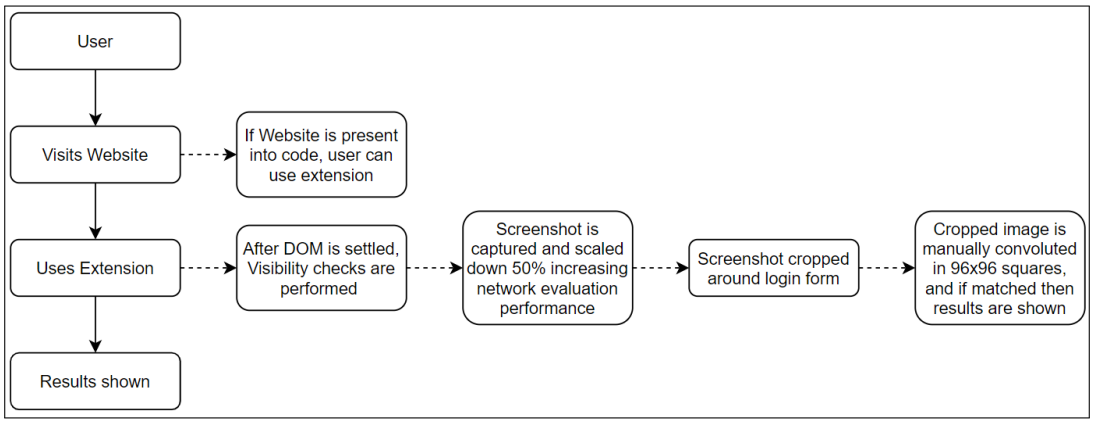

Figure 4: User Workflow

User has the ability to utilize the extension with the help of browser. The proposed system works on domains which are manually coded inside the Neural Network which lowers the chance of visiting any similar phishing Web Page. After successful DOM settlement, the extension then performs visibility check i.e. presence of 'Username' and 'Password' field into the Web Page. If all the prerequisite checks are fulfilled then a screenshot is captured which then gets scaled down to 50% to reduce size and subsequently increasing performance of network evaluation. The screenshot is then cropped around the login form and converted into 96x96 squares. These cropped images are then manually convoluted and matched against our trained artefact and result is shown to the user.

5 Implementation

In this section of report, we will look at the aspect of implementation with respect to our artefact and we will focus on the description of developed solution. The implementation of proposed system is broadly divided into three phases:

5.1 Data Collection

The artefact is based on image similarity check and it is a trivial task to get the image samples for training Neural Network. These images are gathered manually by visiting websites and they are stored into respective folders inside "Sample" as their parent folder. In our proposed system, we've taken our sample dataset as selective 30 websites from top-100 visited websites in the world [52].

5.2 Neural Network Training

Collection of dataset follows training of Neural Network, for which a configuration file is required. This file is in form of ". json" and it contains the Domain and ID of Web Page which is separate for every website. We've configured a JavaScript file to train the Neural Network, which contains the vital part of training. To train our artefact, we will run a shell script which invokes trainer and the Neural Network starts to train. Here, the samples are sliced and broken down into 96x96 squares and then the image is passed through three alternative layers of convolution and pooling which follows a softmax layer to compensate for the regression or loss caused in this process. This process keeps on repeating every 1000 ticks and is halted when the desired accuracy is reached. Implementation of this can be seen in the below Figure 5.

```
const SLICE_SIZE = 96;
const MAX_NEGATIVE_RATIO = 0.30;
const INCREASE_NEGATIVE_TRAINING_THRESHOLD = 0.80;
const TARGET_ACCURACY = 0.99
var labels= common.get_all_labels();
var layer_defs = [];
layer_defs.push({type:'input', out_sx:SLICE_SIZE, out_sy:SLICE_SIZE, out_depth:3});
layer_defs.push({type:'conv', sx:5, filters:18, stride:1, pad: 2, activation:'relu'});
layer_defs.push({type:'pool', sx:4, stride:2});
layer_defs.push({type:'conv', sx:5, filters:20, stride:1, pad: 2, activation:'relu'});
layer_defs.push({type:'pool', sx:4, stride:2});
layer_defs.push({type:'conv', sx:5, filters:20, stride:1, pad: 2, activation:'relu'});
layer_defs.push({type:'pool', sx:4, stride:2});

layer_defs.push({type:'softmax', num_classes:labels.length});
```

Figure 5: Implementation of CNN

5.3 Usability

Once user imports our artefact into their *Google-Chrome* browser and visits any login page, the developed solution then takes a screenshot of it, scales it down to 50% and crops it around login form. This is then divided into 96x96 squares and the image is ran through system followed by performing similarity checks by the Convolutional Neural Network. If the page is phishing, the artefact will warn the user about it and if the visited page resembles any page in our dataset, then this will be prompted as well.

6 Evaluation

In this section we will be providing findings of our study and a comprehensive analysis of results. The Convolutional Neural Network is an advanced level Deep Learning algorithm and it has variety of measurements like Accuracy, Precision and F1-Score with confusion matrix. However, the developed solution discussed in this paper is detecting phishing Web Pages and since measurements are taken manually, the most effective and efficient way to determine accuracy of this model is to calculate True Positive, False Positive and False Negative values of every page. We have used the following set of values in order to analyse the performance of our model:

1. TP (True Positive) - Represents the number of true positive classes, i.e. the number of samples predicted *correctly* (Correct Detection) by the model.

2. FP (False Positive) – Represents the number of false positive classes, i.e. the number of samples predicted *incorrectly* (Incorrect Detection) by the model

3. FN (False Negative) - Representing the number of false negative classes. i.e. the number of samples predicted *incorrectly* (Missed Detection) by the model

6.1 Experiment

6.1.1 Data Pre-processing

The dataset in our artefact is made up of several login Web Page images which are used to train our CNN model. After importing our developed solution in the *Google-Chrome* browser, the images are gathered manually and collected, which are later used to train the Neural Network. All the pre-processing is done via the extension including reduction of the image to 50% and cropping near the login page but storing the image in respective folder is completed by the 'Admin'.

6.1.2 Experimental Settings

The experimental setting covers every aspect of the setup of both the experiments. Firstly, the experiment has been conducted on 15 Web Pages and later 30 Web Pages. To understand the relationship between sample size and accuracy, these different measures are taken. The phishing detection in our experiment is purely based on image detection of login Web Pages. The trained Convolutional Neural Network model will be in-charge to identify and detect any phoney Web Pages presented to our artefact.

6.1.3 Experiment One

Firstly, we've collected 15 Web Page login images from our artefact and these images are used to train the Neural Network. Once the Neural Network is trained, we've tested the websites manually and the results are illustrated in the Table 1 below. After the manual test one, we've gotten 96% of True Positive rate from our model.

Table 1: Experiment One Result Analysis

Total Pages Tested	**True Positive (Correct Detection)**	**False Positive (Incorrect Detection)**	**False Negative (Missed detection)**
15	14(96%)	1(4%)	0(0%)

6.1.4 Experiment Two

In the second phase of test, we've added more 15 Web Page login images in the Neural Network and trained it. Once it has completed training, we've visited every page again and tested the pages manually. As the Table 2 suggests, we've again got a True Positive rate of 96%.

Table 2: Experiment Two Result Analysis

Total Pages Tested	**True Positive (Correct Detection)**	**False Positive (Incorrect Detection)**	**False Negative (Missed detection)**
30	29(96%)	1(4%)	0(0%)

6.2 Discussion

The aim of this study was to identify phishing Web Pages using Convolutional Neural Network and ADADELTA. The findings in our experimental results suggests that this technique is acceptable for detection of phishing. The reason to perform two different set of experiment on the model was to test the consequence of introducing more sample into our training model. After performing the first experiment with 15 Web Pages, it was evident that model was performing well but training phase of our model was time consuming. To check our hypothesis of sample size consequence, experiment two was conducted with 30 Web Pages which performed correctly yielding in 96% of accurate prediction. These results should be taken into account when considering future scope of the developed solution.

These experiments and their consequent results provide clearer understanding in the methodology and scope of Convolutional Neural Network and the results are in-line with our hypothesis. However, time taken to train the proposed solution of Convolutional Neural Network is high and it has a room to improve. While previous researches were focused on detecting handwriting or image classification or identifying non-halal foods using CNN and other algorithms, our results demonstrated that Convolutional Neural Network along with ADADELTA helps to detect phishing websites [56] [57] [58]. Analysing previous researches resulted in immense knowledge and clarity of the CNN which helped in making our artefact better. The vital finding of this paper consists of the usability and flexibility of CNN on phishing detection by a mode of visual similarity. The experiment conducted in this research paper provides a new insight into the vast array of possible endeavours with Convolutional Neural Network.

The developed solution poses some limitations with respect to training period, image input and URL scan amongst other things. The image sample-set is required to be captured from our artefact, no other form of image is acceptable such as a screenshot of a login Web Page. If a masked URL Web Page, identical to the legitimate Web Page is shown to the artefact then the artefact has higher probability of misjudging the result. Time taken to train the proposed solution of Convolutional Neural Network is high and it has a room to improve. Also, due to lack of sample size, the generalizability of the result is limited.

7 Conclusion and Future Work

The unawareness and novice nature of internet users are major factors in case of successful phishing attacks. The adaptiveness and economic motivation make phishers dangerous to users and it is a need of time to pursue different approach to mitigate phishing attacks. This study proposes a visual similarity-based detection scheme and explores the possibility of detecting phishing Web Pages by using two techniques, Convolutional Neural Network and ADADELTA as a combined mechanism. Artefact proposed in this study is

an extension for *Google-Chrome* browser. To evaluate the proposed developed solution, several login Web Page images were personally collected from legitimate websites and used to train the Neural Network. Based on the acquired result we can deduce that distinguishing websites from image pattern and login form along with cross-verifying URL is an effective and efficient way of identifying a phishing Web Page.

In addition, the research revealed some features and flaws of Convolutional Neural Network. On one hand, CNN has capability to identify even miniscule variation in images but the training phase of Neural Network is time-consuming. While carrying out extensive experimental training and analysis, the developed artefact showed dependency of sample size on training time and hence it can vary if larger sample-set is introduced. In foreseeable future, more research can be emphasized in focusing on polishing the methodology along with integrating *Keras* and optimized GPU acceleration to improve training time. The training period can also be improved by using different set of tools and/or platform to train and implement the artefact. An improvement in the training period can also be highly beneficial to the user since the model can learn at high rate reducing the training period and improving overall usability of the artefact. In addition to *Google-Chrome*, there is a window for our extension for broadening the horizon to other web browsers and securing more users. As the technology evolves, a continuous research is necessary to mitigate and eradicate any attempts of phishing.

References

[1] KnowBe4, "Phishing | History of Phishing," library Catalog: www.phishing.org. [Online]. Available: https://www.phishing.org/history-of-phishing

[2] "Gartner Forecasts Worldwide Information Security Spending to Exceed $124 Billion in 2019," library Catalog: www.gartner.com. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-billion-in-2019

[3] "Global Cybersecurity Spending Predicted To Exceed $1 Trillion From 2017-2021," Jun. 2019, library Catalog: cybersecurityventures.com Section: Cybersecurity Market Report. [Online]. Available: https://cybersecurityventures.com/cybersecurity-market-report/

[4] "Mobile Threat Landscape Report 2020," library Catalog: www.wandera.com. [Online]. Available: https://www.wandera.com/mobile-threat-landscape/

[5] APWG, "Phishing activity trends report." [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q4_2019.pdf

[6] "2019 Internet Crime Report Released," library Catalog: www.fbi.gov. [Online]. Available: https://www.fbi.gov/news/stories/2019-internet-crime-report-released-021120

[7] M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013, conference Name: IEEE Communications Surveys Tutorials.

[8] G. Xiang, A. Hauptmann, C. Faloutsos, and M. Jakobsson, *Toward a Phish Free World: A Feature-type-aware Cascaded Learning Framework for Phish Detection.*

[9] "Smartening the crowds | Proceedings of the Seventh Symposium on Usable Privacy and Security." [Online]. Available: https://dl.acm.org/doi/10.1145/2078827.2078838

[10] "Teaching Johnny not to fall for phish | ACM Transactions on Internet Technology." [Online]. Available: https://dl.acm.org/doi/10.1145/1754393.1754396

[11] M. He, S.-J. Horng, P. Fan, M. K. Khan, R.-S. Run, J.-L. Lai, R.-J. Chen, and A. Sutanto, "An efficient phishing webpage detector," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 018–12 027, Sep. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411000662

[12] T. Li, F. Han, S. Ding, and Z. Chen, "LARX: Large-Scale Anti-Phishing by Retrospective Data-Exploring Based on a Cloud Computing Platform," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, Jul. 2011, pp. 1–5, iSSN: 1095-2055.

[13] "A multi-tier phishing detection and filtering approach | Journal of Network and Computer Applications." [Online]. Available: https://dl.acm.org/doi/10.1016/j.jnca.2012.05.009

[14] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 301–311, Oct. 2006, conference Name: IEEE Transactions on Dependable and Secure Computing.

[15] K.-T. Chen, J.-Y. Chen, C.-R. Huang, and C.-S. Chen, "Fighting Phishing with Discriminative Keypoint Features," *IEEE Internet Computing*, vol. 13, no. 3, pp. 56–63, May 2009, conference Name: IEEE Internet Computing.

[16] T. Raffetseder, E. Kirda, and C. Kruegel, "Building Anti-Phishing Browser Plug-Ins: An Experience Report," in *Third International Workshop on Software Engineering for Secure Systems (SESS'07: ICSE Workshops 2007)*, May 2007, pp. 6–6.

[17] C. Yue and H. Wang, "Anti-Phishing in Offense and Defense," in *2008 Annual Computer Security Applications Conference (ACSAC)*, Dec. 2008, pp. 345–354, iSSN: 1063-9527.

[18] Y. Joshi, S. Saklikar, D. Das, and S. Saha, "PhishGuard: A browser plug-in for protection from phishing," in *2008 2nd International Conference on Internet Multimedia Services Architecture and Applications*, Dec. 2008, pp. 1–6.

[19] N. Mazher, I. Ashraf, and A. Altaf, "Which web browser work best for detecting phishing," in *2013 5th International Conference on Information and Communication Technologies*, Dec. 2013, pp. 1–5.

[20] G. Varshney, M. Misra, and P. Atrey, "Browshing a new way of phishing using a malicious browser extension," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Apr. 2017, pp. 1–5.

[21] G. Varshney, S. Bagade, and S. Sinha, "Malicious browser extensions: A growing threat: A case study on Google Chrome: Ongoing work in progress," in *2018 International Conference on Information Networking (ICOIN)*, Jan. 2018, pp. 188–193.

[22] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th international conference on World Wide Web*, ser. WWW '07. Banff, Alberta, Canada: Association for Computing Machinery, May 2007, pp. 639–648. [Online]. Available: https://doi.org/10.1145/1242572.1242659

[23] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites," *ACM Transactions on Information and System Security*, vol. 14, no. 2, pp. 21:1–21:28, Sep. 2011. [Online]. Available: https://doi.org/10.1145/2019599.2019606

[24] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, "Client-side defense against web-based identity theft," p. 16.

[25] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," in *Special interest tracks and posters of the 14th international conference on World Wide Web*, ser. WWW '05. Chiba, Japan: Association for Computing Machinery, May 2005, pp. 1060–1061. [Online]. Available: https://doi.org/10.1145/1062745.1062868

[26] "(PDF) An antiphishing strategy based on visual similarity assessment," library Catalog: www.researchgate.net. [Online]. Available: https://www.researchgate.net/publication/3419805_An_antiphishing_strategy_based_on_visual_similarity_assessment#fullTextFileContent

[27] Y. Liu, W. Liu, and C. Jiang, "User Interest Detection on Web Pages for Building Personalized Information Agent," in *Advances in Web-Age Information Management*, ser. Lecture Notes in Computer Science, Q. Li, G. Wang, and L. Feng, Eds. Berlin, Heidelberg: Springer, 2004, pp. 280–290.

[28] "(PDF) EMD based Visual Similarity for Detection of Phishing Webpages," library Catalog: www.researchgate.net. [Online]. Available: https://www.researchgate.net/publication/228846659_EMD_based_Visual_Similarity_for_Detection_of_Phishing_Webpages

[29] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, ser. SecureComm '08. Istanbul, Turkey: Association for Computing Machinery, Sep. 2008, pp. 1–6. [Online]. Available: https://doi.org/10.1145/1460877.1460905

[30] G. Xiang, A. Hauptmann, C. Faloutsos, and M. Jakobsson, *Toward a Phish Free World: A Feature-type-aware Cascaded Learning Framework for Phish Detection.*

[31] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[33] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[34] M. J. L. Orr, "Centre for Cognitive Science, University of Edinburgh, 2, Buccleuch Place, Edinburgh EH8 9LW, Scotland April 1996," p. 67.

[35] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980. [Online]. Available: https://doi.org/10.1007/BF00344251

[36] K. Fukushima and S. Miyake, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition," in *Competition and Cooperation in Neural Nets*, ser. Lecture Notes in Biomathematics, S.-i. Amari and M. A. Arbib, Eds. Berlin, Heidelberg: Springer, 1982, pp. 267–285.

[37] D. E. Rumelhart and J. L. McClelland, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. MITP, 1987, pp. 318–362, conference Name: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations. [Online]. Available: https://ieeexplore.ieee.org/document/6302929

[38] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, conference Name: Neural Computation.

[39] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, conference Name: Proceedings of the IEEE.

[40] L. Chen, C. Wu, W. Fan, J. Sun, and S. Naoi, "Adaptive Local Receptive Field Convolutional Neural Networks for Handwritten Chinese Character Recognition," in *Pattern Recognition*, ser. Communications in Computer and Information Science, S. Li, C. Liu, and Y. Wang, Eds. Berlin, Heidelberg: Springer, 2014, pp. 455–463.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[42] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, iSSN: 1063-6919.

[43] "(PDF) Hand Posture Recognition Using Convolutional Neural Network," library Catalog: www.researchgate.net. [Online]. Available: https://www.researchgate.net/publication/329374701_Hand_Posture_Recognition_Using_Convolutional_Neural_Network

[44] N. Alnaim, M. Abbod, and A. Albar, "Hand Gesture Recognition Using Convolutional Neural Network for People Who Have Experienced A Stroke," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Oct. 2019, pp. 1–6.

[45] Z. Hu, I. Tereykovskiy, Y. Zorin, L. Tereykovska, and A. Zhibek, "Optimization of Convolutional Neural Network Structure for Biometric Authentication by Face Geometry," in *Advances in Computer Science for Engineering and Education*, ser. Advances in Intelligent Systems and Computing, Z. Hu, S. Petoukhov, I. Dychka, and M. He, Eds. Cham: Springer International Publishing, 2019, pp. 567–577.

[46] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv:1212.5701 [cs]*, Dec. 2012, arXiv: 1212.5701. [Online]. Available: http://arxiv.org/abs/1212.5701

[47] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," p. 39.

[48] "(PDF) Improving the Convergence of Back-Propagation Learning with Second-Order Methods," library Catalog: www.researchgate.net. [Online]. Available: https://www.researchgate.net/publication/216792889_Improving_the_Convergence_of_Back-Propagation_Learning_with_Second-Order_Methods

[49] "methodology - Research -Methodology," library Catalog: research-methodology.net. [Online]. Available: https://research-methodology.net/research-methodology/

[50] "(PDF) Secure Spiral: A Secure Software Development Model," library Catalog: www.researchgate.net. [Online]. Available: https://www.researchgate.net/publication/272953501_Secure_Spiral_A_Secure_Software_Development_Model

[51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, number: 6088 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/323533a0

[52] J. Hardwick, "Top 100 Most Visited Websites by Search Traffic (as of 2020)," Jan. 2020, library Catalog: ahrefs.com. [Online]. Available: https://ahrefs.com/blog/most-visited-websites/

[53] "ConvNetJS: Deep Learning in your browser." [Online]. Available: https://cs.stanford.edu/people/karpathy/convnetjs/index.html

[54] "OWASP Secure Coding Practices-Quick Reference Guide," library Catalog: owasp.org. [Online]. Available: https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/migrated_content.html

[55] "protecting users importance defending-public-sites." [Online]. Available: https://software-security.sans.org/resources/paper/reading-room/protecting-users-importance-defending-public-sites

[56] K. Chen, K. Zhu, and M. Meng, "Image-enhanced Adaptive Learning Rate Handwritten Vision Processing Algorithm Based on CNN," in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, Jul. 2019, pp. 112–116.

[57] Y. Luan and S. Lin, "Research on Text Classification Based on CNN and LSTM," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Mar. 2019, pp. 352–355.

[58] H. Fadhilah, E. C. Djamal, R. Ilyas, and A. Najmurrokhman, "Non-Halal Ingredients Detection of Food Packaging Image Using Convolutional Neural Networks," in *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, Aug. 2018, pp. 131–136.