ISSN-2347-2227
Subscribers copy
Not for sale

i-manager's

Journal on Computer Science

Disseminating new ideas in Information and Computation

i-manager®
Publications

i-manager Publications

ISSN-2347-2227
Volume 6. No. 3
September - November 2018

i-manager's

Journal on Computer Science

About the Journal

i-manager's Journal on Computer Science deals with all aspects of computer science and contributes theoretical results and offers a compilation of high quality articles to encompass a wide spectrum of advancements in the actively developed domain. i-manager's Journal on Computer Science covers a great deal of what has been done in the field recently and intends to bring together the most recent advances and applications in all branches of the academic computer science community with new knowledge and technology for the benefit of students, professionals and industrial practitioners.

i-manager's Journal on Computer Science is presently in its 6^{th} Year. The first issue was launched in 2013.

i-manager's Journal on Computer Science is published by i-manager Publications, one of India's leading Academic Journal Publisher, publishing 28 Academic Journals in diverse fields of Engineering, Education, Management and Science.

Why Publish with us

i-manager Publications currently publishes academic Journals in Education, Engineering, Scientific and Management streams. All of i-manager's Journals are supported by highly qualified Editorial Board members who help in presenting high quality content issue after issue. We follow stringent Double Blind Peer Review process to maintain the high quality of our Journals. Our Journals target both Indian as well as International researchers and serve as a medium for knowledge transfer between the developed and developing countries. The Journals have a good mix of International and Indian academic contributions, with the peer-review committee set up with International Educators.

Submission Procedure

Researchers and practitioners are invited to submit an abstract of maximum 200 words on or before the stipulated deadline, along with a one page proposal, including Title of the paper, author name, job title, organization/institution and biographical note.

Authors of accepted proposals will be notified about the status of their proposals before the stipulated deadline. All submitted articles in full text are expected to be submitted before the stipulated deadline, along with an acknowledgement stating that it is an original contribution.

Review Procedure

All submissions will undergo an abstract review and a double blind review on the full papers. The abstracts would be reviewed initially and the acceptance and rejection of the abstracts would be notified to the corresponding authors. Once the authors submit the full papers in accordance to the suggestions in the abstract review report, the papers would be forwarded for final review. The final selection of the papers would be based on the report of the review panel members.

Format for Citing Papers

Author surname, initials (s.) (2018). Title of paper, i-manager's Journal on Computer Science, 6(3), xx-xx.

Copyright

Contact e-mails

editor_jcom@imanagerpublications.com
submissions@imanagerpublications.com

ISSN-2347-2227
Volume 6. No. 3
September - November 2018

i-manager's

Journal on Computer Science

i-manager Publications

ISSN-2347-2227
Volume 6. No. 3
September - November 2018

i-manager's
Journal on Computer Science

OUR OFFICES

Registered Office

3/343,Hill view,
Town Railway Nager,
Nagercoil, Kanyakumari District - 629001
Ph : 91-4652- 277675
E-mail : info@imanagerpublications.com

Editorial Office

13-B, Popular Building,
Mead Street, College Road,
Nagercoil, Kanyakumari District - 629001
Ph : (91-4652) 231675, 232675, 276675
E-mail : editor_jcom@imanagerpublications.com

Abstracting / Indexing

ProQuest ULRICHSWEB GLOBAL SERIALS DIRECTORY Google Scholar OCLC WorldCat

Join with us

https://www.facebook.com/Journal-on-Computer-Science-2168468313379110/

https://www.facebook.com/imanagerPublishing/

https://twitter.com/imanagerpub

CONTENTS

EDITORIAL

The current issue of i-manager's Journal on Computer Science mainly focuses on Authentication System, Artificial Neural Network used to detect e-banking Phishing Websites, Password Management, Adaptive Personnel Selection Expert System to Support Organization's Personnel Recruitment Decision Process, Evaluation of Classification Algorithms for Phishing URL Detection and detection of Captcha Smuggling Attacks using Supervised Deep Learning Based Approach.

Omorogiuwa and his co-author Aziken have proposed a study about Computer-Based Local Area Authentication System. The system was developed using XAMPP integrated net-base application and JAVA object-oriented programming language. This security system is controlled through the network via the server and controls all clients that choose to use the resources like e-exam platform, e-library, etc. The performance of the system has been monitored and the result is found to be satisfactory, as all unauthorized users are blocked and appropriate warning messages are sent to the client's system by the server when the user attempts to login which eliminates external users from gaining access to the examination platform.

Shafi'i Muhammad Abdulhamid et al., have proposed a study about a soft computing approach to detect e-banking phishing websites using Artificial Neural Network (ANN). Confusion matrix analysis was used in this study to detect e-banking phishing websites. Datasets from various websites comprises of both legitimate and phishing websites collected from directory and analysed by ANN Algorithm with Confusion Matrix. The study results showed that the proposed ANN algorithm produces a remarkable percentage of accuracy and reduced false positive rate during detection and can produce competitive results that is suitable for detecting phishing in e-banking websites.

Victor N. Adama et al., have presented a study to analyse about password knowledge and password management. This research was conducted via a case study aimed at establishing the theoretical password knowledge in comparison to actual password management practice of staff and students from Information Technology (IT) inclined departments of the Federal University of Technology, Minna. The data collection was carried out primarily based on a survey. The study results concluded that, there is a significant difference between what respondents know compared to their actual practice. The authors recommend that, more extensive research into enhancing graphical password entropies are to be conducted in future as they possess the potential to replace text passwords.

Muhammad Ahmad Shehu et al., have proposed a study to analyze the personnel recruitment operation which is an essential human resource operation of an organization. An adaptive personnel selection model was developed to minimize the complexity and to carry out the personnel selection by considering some of the operational behaviors. The adaptive personnel selection model was developed using a C4.5 decision tree and frequent and non-frequent pattern analysis of data mining. The study results showed that, the proposed expert system enables the personnel selection strategy changes to be fed in by the organization, when it occurs.

Oluyomi Ayanfeoluwa et al., have conducted a study to evaluate the capacity of different algorithms to detect phishing URLs. Dataset was obtained from UCI Machine Learning Repository, and the algorithms were assessed in terms of Accuracy, Precision, Recall, F-Measure, Receiver Operating Characteristic (ROC) area and Root Mean Squared Error (RMSE). In terms of accuracy, precision, recall, F-measure, and RMSE, the Random Forest algorithm was found to perform better than the other algorithms analyzed and a number of others from existing literature. The authors recommend that, further studies are to be conducted, to ascertain if performances are dataset-specific.

Moses O. Omoyele et al., have proposed a study to analyze a predictive model for the detection of captcha smuggling attacks. In order to achieve the aim, framework based on hyper parameter specification was developed in this study. The model was evaluated on the available CAPTCHA smuggling dataset. The outcome of this research will benefit web developers, web users, web hosting companies and internet service providers. The study results showed that, the accuracy of prediction achieved in this work is 77.89% at consistency of 0.1543. The sensitivity and specificity of the model are 78.11% and 78.2%, respectively.

All papers of this issue, papers 1 to 6 were submitted from the 2nd International Conference on Information and

EDITORIAL

Communication Technology and Its Applications (ICTA 2018), conducted on 5 -6th September 2018 at Federal University of Technology, Minna, Nigeria. We express our gratitude to the Conveners Dr. Shafii Abdulhamid & Dr. Oluwafemi Osho for their support in ensuring the papers were submitted on time.

We extend our sincere thanks to the authors for their contributions towards this issue and we are grateful to the reviewers for spending their quality time in reviewing these papers. Our special thanks to the Editor-in-Chief, Dr. Kamal kumar Mehta for his continuous support and efforts in improving further the quality of the Journal.

Enjoy reading!

Warm regards,

Ramani R
Junior Associate Editor
i-manager Publications

DEVELOPMENT OF A PREDICTIVE MODEL FOR THE DETECTION OF CAPTCHA SMUGGLING ATTACKS USING SUPERVISED DEEP LEARNING BASED APPROACH

By

MOSES O. OMOYELE * JOSEPH A. OJENIYI ** OLAWALE S. ADEBAYO ***

* Research Scholar, Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria.
-* Lecturer, Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria.

Date Received: 11/01/2019 Date Revised: 28/01/2019 Date Accepted: 22/03/2019

ABSTRACT

CAPTCHA is a piece of program designed to distinguish human beings from bots. These are computer generated tests which can be solved by humans but will be difficult to be solved by computers. Bots smuggled CAPTCHAs are gradually on the increase in order to deceive unsuspecting users and inadvertently infect systems. From the available literature reviewed so far, there is no model to detect or predict CAPTCHA smuggling attack. The aim of this work is to come up with a model capable of predicting this attack. The approach used was based on deep supervised neural network approach. In order to achieve the aim, framework based on hyperparameter specification was developed. The model was evaluated on the available CAPTCHA smuggling dataset. The accuracy of prediction achieved in this work is 77.89% at consistency of 0.1543. The sensitivity and specificity of the model are 78.11% and 78.2%, respectively.

Keywords: CAPTCHA, CAPTCHA Smuggling, Deep Learning Model.

INTRODUCTION

Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a multimedia security mechanism also referred to as Human Interactive Proofs (HIP) (Bilge, Strufe, Balzarotti, & Kirda, 2009; Chen, Luo, Guo, Zhang, & Gong, 2017). CAPTCHA has the ability to enhance the privacy of multimedia. Its successful deployment and applications have been recorded in Yahoo, Google, Microsoft, and several other major websites. The efforts in breaking CAPTCHA came into being in order to verify reliability, robustness and security of the CAPTCHA. Image processing, pattern recognition, artificial intelligence and computer vision are major technologies involved in this CAPTCHA breaking attempts. The research on CAPTCHA breaking has great value in research and application. CAPTCHA is an integral part of artificial intelligence and an important prerequisite to actualize natural human-computer interaction.

CAPTCHA was first introduced by Von Ahn et al. in the year 2003 (Von Ahn, Blum, Hopper, Langford, 2003; Gupta & Garg, 2015). The work was later elaborated to include different techniques that can be used to tell computers and humans apart automatically (Chen, Luo, Guo, Zhang, & Gong, 2017; Von Ahn, Blum, & Langford, 2004). Image recognition based challenges were the focus of Chew et al. (Chew & Tygar, 2004; Sivakorn, Polakis, & Keromytis, 2016a). A further system proposed enables the human user to describe the subject in a picture, or recognize an interfering image from an otherwise coherent set of pictures. Making CAPTCHAs usable on mobile devices is the further contribution of by Chow et al. in the year 2008. The system does not rely on keyboard input, which can be annoying especially on mobile devices. Instead, they designed a CAPTCHA that can be solved with touch screens or numeric keypads (Chow, Golle, Jakobsson, Wang & Wang, 2008; Alsuhibany, 2016; Hernández-Castro, R-Moreno, Barrero, & Gibson, 2017).

In the work of (Egele, Bilge, Kirda, & Kruegel, 2010; Sharma & Seth, 2015; Chilluru, Naick, & Nirupama, 2015), a novel attack denoted as CAPTCHA smuggling was presented. In a CAPTCHA smuggling attack, user interactions with legitimate online services (such as web mail or social networking sites) are intercepted by the attacker (i.e., a malicious program executing on the victim's computer) and put on hold until the victim solves a CAPTCHA challenge. The displayed CAPTCHA and its surrounding browser window spoof the visual characteristics of the online service that the victim is using. Hence, it is difficult for victims to distinguish between real CAPTCHAs displayed by the online service and CAPTCHAs smuggled into the session by the attacker. As the CAPTCHA challenge is under the direct control of the attacker, a malicious program that needs to solve a CAPTCHA can forward the challenge to a victim's computer. The malicious component on this computer then performs the CAPTCHA smuggling attack (and thus, gets the challenge solved by the unsuspecting user). The premise of the attack is that users are so accustomed to solving CAPTCHAs while using online services that they will not notice extra CAPTCHAs that are smuggled in by a malicious application running on their computer (Egele, Bilge, & Kirda, Kruegel, 2010; Uzun, Chung, Essa, & Lee, 2018; Nguyen, Chow, & Susilo, 2014).

The typical attack scenario presented by Egele, Bilge, Kirda, and Kruegel, 2010 involves a botnet with bots that intercept user interactions and smuggle CAPTCHAs into the victim's active web browsing sessions. For example, a Facebook CAPTCHA that is under the attacker's control would sometimes be displayed when the victim starts to compose a message or send a friend request (Sivakorn, Polakis, & Keromytis, 2016a; Sivakorn, Polakis, & Keromytis, 2016b). Requiring a victim to solve only a few CAPTCHAs a day ensures that the manipulation stays unnoticed and is perceived as normal procedure. Note that a CAPTCHA smuggling attack is very lightweight in terms of required resources. Therefore, it is trivial for the bot master to add the required functionality to the existing bot program without limiting the existing functionality of the botnet.

1. Problem Statement

CAPTCHA is a piece of program designed to distinguish human beings from bots. These are computer generated tests which can be solved by humans but will be difficult to be solved by computers. Bots smuggled CAPTCHAs are gradually on the increase in order to deceive unsuspecting users and inadvertently infect systems. There is a need to come up with an approach to detect and mitigate this ugly incident. Development of a predictive model will ensure that these smuggled CAPTCHAs are not only prevented but also denied access to the system. The focus of this paper is to develop a predictive model for detecting CAPTCHA Smuggling Attacks using supervised deep learning approach.

2. Methods

2.1 Research Design

The design of this work follows the block diagram in Figure 1. The first block focuses on development of hyperparametric framework for the proposed model. This was followed by multi-layer network design and development. The network nodes were connected. The developed network model was subsequently trained and validated. In order to improve the accuracy of prediction, dynamic thresholding was employed. The model was then tested and evaluated.

2.2 Multi-Layered Hyperparametric Framework for the Proposed Model

The hyperparametric framework was designed based on the multi-layered perspectives. Three layers were used in the framework. Two of the layers are hidden while there is one layer in the output section. The neurons in both hidden

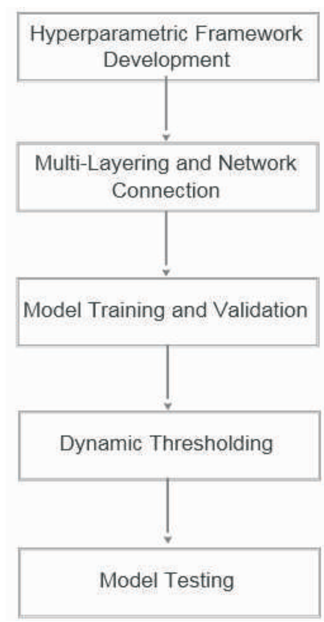

Figure 1. Research Design Block Diagram

and output layers were iterated in order to perform deep learning of the training dataset as shown in Figure 2.

2.2.1 Parameter Specification

2.2.1.1 Input Parameter

Normalized Browser Activity (NBA): The input parameter is fed into a neuron in ANN-based model as in Figure 3.

2.2.1.2 Weight Parameters

Connection Weights (w_{kj}, v_{kj}): there are two categories of weights – Input Weights and Layer Weights.

Input Weights: The connections between the input nodes and the hidden layer are associated with weights denoted by w_{kj}.

Layer Weights: The notation (v_{kj}) is used to denote connections between the hidden layer and the output layer. Both weights are proportional to the number of neurons in the hidden and output layers. The weight parameters are initialized as shown in Figure 4.

2.2.1.3 Summing Function Parameters

Adder (A): This is an adder of the weighted input values. It exists in each neuron both in the hidden layer and output layer. The processes of the adder are depicted in Figure 5.

2.2.1.4 Activation Function Parameters

Transfer (I): This is a normalizing function that normalizes the summed weighted inputs into the model output. This is depicted in Figure 6.

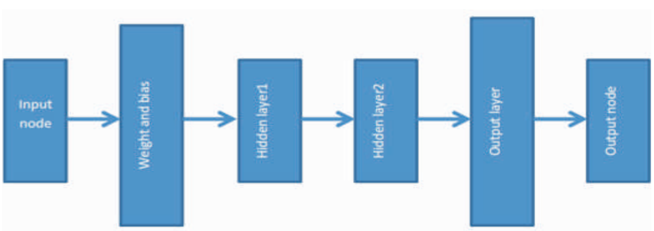

Figure 2. Multi-Layered Hyperparametric Framework Design

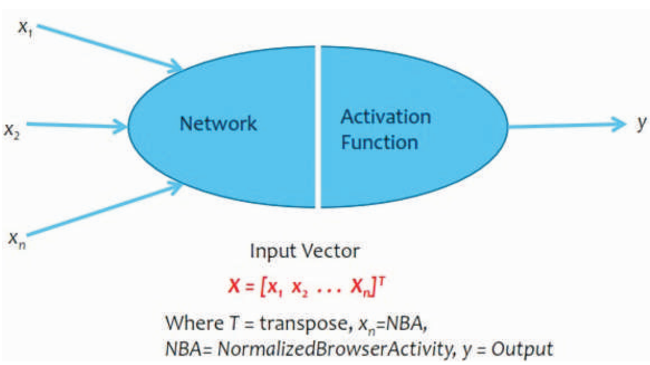

Figure 3. Input Parameters to a Neuron

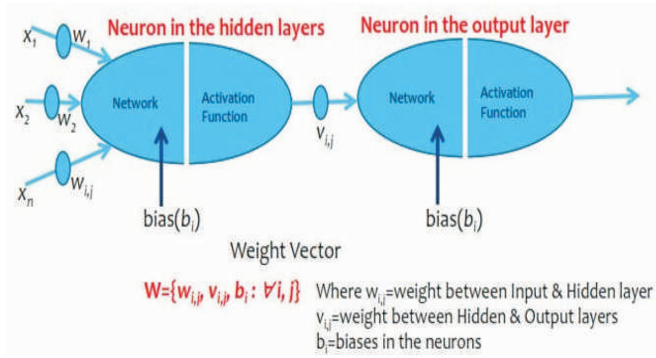

Figure 4. Weight and Bias Initialization

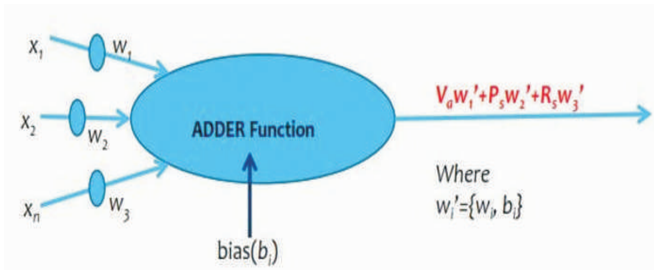

Figure 5. Summing Parameters

2.2.1.5 Error Parameters

The difference between the grand truth values of the output in the dataset and the model output is the error of processing. It is the difference between the target (output from dataset) and output from the developed model. Error parameters contribute largely the number of iterations in the processing of the model. Error parameters are depicted as in Figure 7.

2.2.1.6 Error Back Propagation Parameters

As much as possible, the model intelligently keeps errors at minimal values through backward propagation of

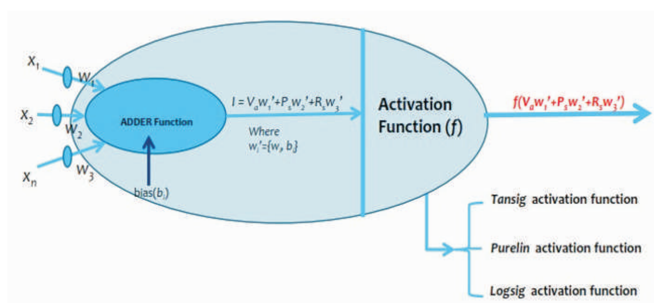

Figure 6. Activation Function Parameters

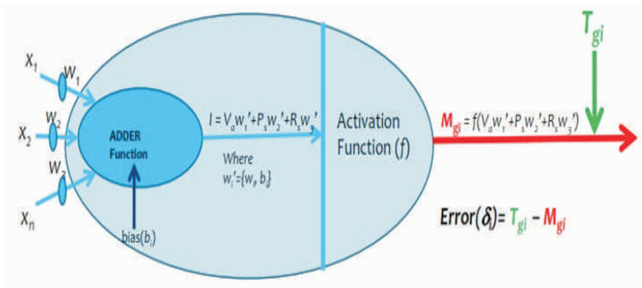

Figure 7. Error Parameters

errors to the processing units based on the ratio of weights and biases. The back propagated errors are readjusted and fed forward for subsequent iteration. These procedures are captured in Figure 8.

2.2.1.7 Weight and Bias Update Equation

Based on the changes in weights and biases being computed during the backward propagation and feedforward operations of the model, the corresponding equation gets updated. The learning rate, error computations and gradient descent largely constitute in generating the updated equation. The relevant portion of this is captured in Figure 9.

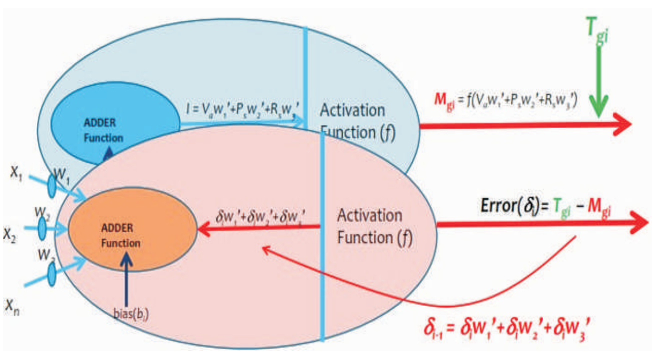

Figure 8. Error Back Propagation Parameters

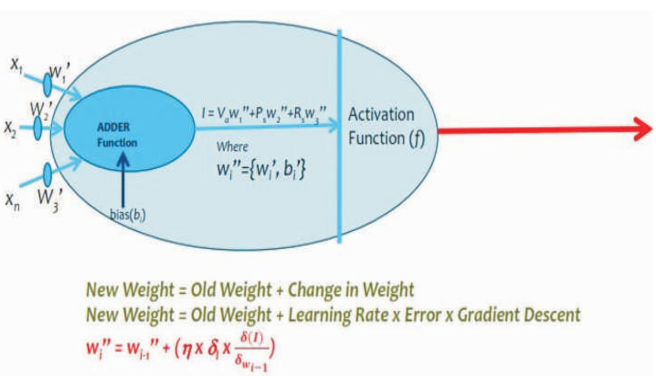

Figure 9. Weight and Bias Update Equation Parameters

2.3 Model Architecture

The architecture of the predictive model is based on artificial neural network (ANN) with two hidden layers and one output layer. Iterative number of neurons were used in the layers based on the hyperparameters. The grand truth values from the dataset referred as the target are also included in the architecture to serve as the training supervisor. The architecture is depicted in Figure 10.

2.4 Developed Mathematical Model

From the model diagram in Figure 10, the input and output matrix equations can be written as:

Input = [NBA] (1)

Where NBA = Normalized Browser Activity

Output = [CSB] (2)

Where CSB=Captcha Smuggling Bit

The equations (1) and (2) can be combined to form linear equation (3)

[CSB]=[Weights] [NBA] (3)

Weights = $[w_{ij}]$ $[v_{ij}]$ (4)

Where w_{ij} are the weights between the input nodes and neurons at the hidden layers and v_{ij} are the weights between the neurons at the hidden layers and the output layers.

The expanded weights equations are given in (5) and (6).

$$w_{ij} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & \cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & w_{24} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & w_{34} & \cdots & w_{3n} \end{bmatrix} \quad (5)$$

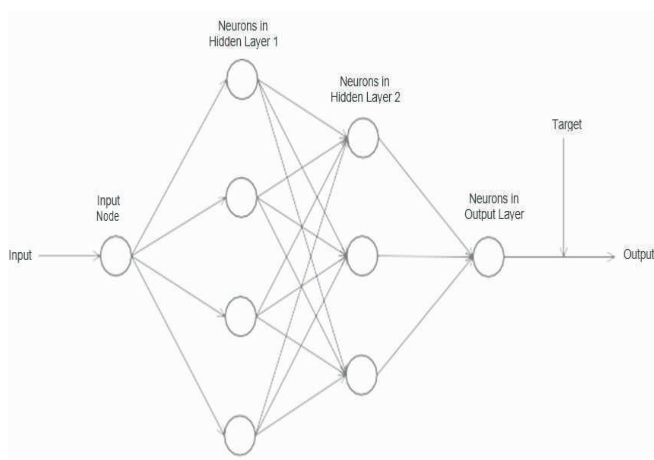

Figure 10. Architecture of the Predictive Model

$$v_{ij} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \\ v_{41} & v_{42} & v_{43} \\ \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & v_{n3} \end{bmatrix} \quad (6)$$

By substituting all equations (4), (5) and (6) into equation (3), it gives equation (7),

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & \cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & w_{24} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & w_{34} & \cdots & w_{3n} \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \\ v_{41} & v_{42} & v_{43} \\ \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & v_{n3} \end{bmatrix} [NBA] \quad (7)$$

$$= \begin{bmatrix} w_{11}v_{11} + w_{12}v_{21} + \ldots & w_{21}v_{11} + w_{22}v_{21} + \ldots & w_{31}v_{11} + w_{32}v_{21} + \ldots \\ w_{11}v_{12} + w_{12}v_{22} + \ldots & w_{21}v_{12} + w_{22}v_{22} + \ldots & w_{31}v_{12} + w_{32}v_{22} + \ldots \\ w_{11}v_{13} + w_{12}v_{23} + \ldots & w_{21}v_{13} + w_{22}v_{23} + \ldots & w_{31}v_{13} + w_{32}v_{23} + \ldots \end{bmatrix} [NBA] \quad (8)$$

The constants (w_{ij} and v_{ij}) coefficients derivable from the ANN-based models developed in Matrix Laboratory (MATLab).

$$\begin{bmatrix} w_{11}v_{11} + w_{12}v_{21} + \ldots & w_{21}v_{11} + w_{22}v_{21} + \ldots & w_{31}v_{11} + w_{32}v_{21} + \ldots \\ w_{11}v_{12} + w_{12}v_{22} + \ldots & w_{21}v_{12} + w_{22}v_{22} + \ldots & w_{31}v_{12} + w_{32}v_{22} + \ldots \\ w_{11}v_{13} + w_{12}v_{23} + \ldots & w_{21}v_{13} + w_{22}v_{23} + \ldots & w_{31}v_{13} + w_{32}v_{23} + \ldots \end{bmatrix}$$

[ANN model weight matrix] (9)

ANN model weight matrix = $[IW_{ij}]' [LW_{ij}]'$ (10)

where IW_{ij} = Input weight matrix, LW_{ij} = Layer weight matrix

The mathematical notation for hidden layer is given as:

$$z_i = x_i \cdot w_{ij} + b_j$$
$$Z = \sum x_i w_{ij} + b_j \quad (11)$$

Where b_j = hidden layer bias

The mathematical notation for output layer is given as:

$$\begin{aligned} y &= z_j v_j + c_j \\ &= v_j (w_{ij} x_i + b_j) + c_j \\ &= LW (IW \cdot X + B_j) + C \end{aligned} \quad (12)$$

The generic model is given as:

$$y = AF_{OL}(LW \cdot AF_{HL}(IW \cdot X + B_j) + C) \quad (13)$$

where AF_{OL} = Activation Function of Output Layer

AF_{HL} = Activation Function of Hidden Layer

X = Input Matrix

LW = Layer Weights Matrix

IW = Input Weights Matrix

B_j = Hidden Layer Bias Matrix

C_j = Output Layer Bias Matrix

The instantiated generic model for Tansig-Purelin Activation Function Combinations is given as:

Recall: The activation function of Tansig is given as:

$$Tansig(x) = \left[\frac{2}{1+e^{-2x}} - 1 \right] \quad (14)$$

$$Tansig(IW \cdot X + B_j) = \left[\frac{2}{1+e^{-2(IW \cdot X + B_j)}} - 1 \right] \quad (15)$$

$$y = Purelin\left(LW \cdot \left[\frac{2}{1+e^{-2(IW \cdot X + B_j)}} - 1 \right] + C \right) \quad (16)$$

The activation function of Purelin is given as:

Purelin (x) = $k_i x_i$ (17)

Purelin (x) = K.X (18)

Therefore, the generic model for Tansig – Purelin Activation Function Combinations is given as:

$$y = K\left(LW \cdot \left[\frac{2}{1+e^{-2(IW \cdot X + B_j)}} - 1 \right] + C \right) \quad (19)$$

3. Results and Discussion

From the developed ANN-based model, the model coefficient in (19) is computed as the product of input weight matrix and layer weight matrix. The coefficient is given in equation (20):

Coefficient (K) = 0.276 (20)

The optimised mathematical model of the developed dynamic thresholding model is given in (21):

$$y = 0.276\left(LW \cdot \left[\frac{2}{1+e^{-2(IWx0.276)}} - 1 \right] + 0.276 \right) \quad (21)$$

3.1 Model Performance Evaluation

The model was evaluated at different threshold refinements. Table 1 shows a refinement results for 0.05 while Figure 11 shows the results for refinement of 0.005. At the refinement of 0.05, the best accuracy attained was 77.89668%. From Figure 11, threshold range from 0.4 to

Threshold Value	Mean Accuracy	Accuracy Deviation
0.1	37.82288	4.027469
0.15	53.39483	7.733247
0.45	77.15867	1.046957
0.5	77.85978	0.130463
0.55	77.89668	0.154365
0.6	76.78967	1.717949
0.65	76.12546	1.606332
0.8	75.38745	1.072653
0.85	74.98155	0.165023
0.9	74.90775	0

Table 1. Model Dynamic Thresholding and Accuracy at Refinement of 0.05

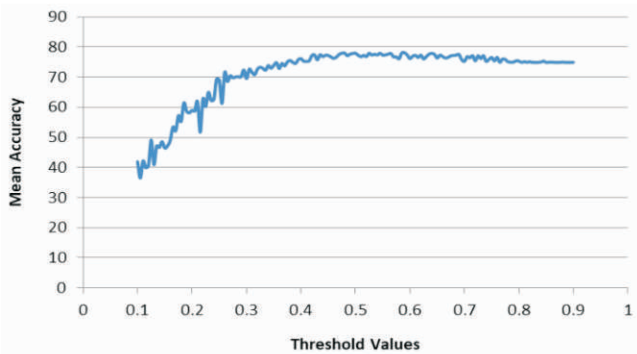

Figure 11. Mean Accuracy Versus Threshold Values at Refinement of 0.005

0.75, falls within the high accuracy region. The threshold value of 0.9 recorded the highest consistency by attaining the least deviation of 0.

At the refinement of 0.005, the accuracy achieved was 78.11808%. At the same refinement, consistency of 0 were achieved over a set of threshold values from 0.805 to 0.9. These are shown in Figures 11 and 12.

The Mean Squared Error of the Model shows 0.19103 within eleven iterations. At 5th epoch, there was convergence as shown in Figure 13. The gradient descent of 0.00064177 with 11 epoch. The predictive model attained its best hyperparametric processing at Tansig activation function in hidden layer 1, Tansig activation function in hidden layer 2 and Purelin Activation function in the output layer. The accuracy-based measures are 78.59% accuracy, 78.4% correct rate, 21.6% error rate, 0% inconclusive rate, 100% classified rate, 78.11% sensitivity and 78.2% specificity as shown in Table 2.

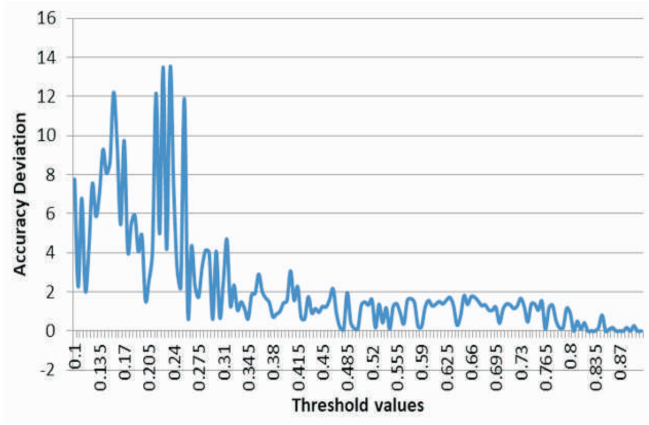

Figure 12. Accuracy Deviation Versus Threshold Values at Refinement of 0.005

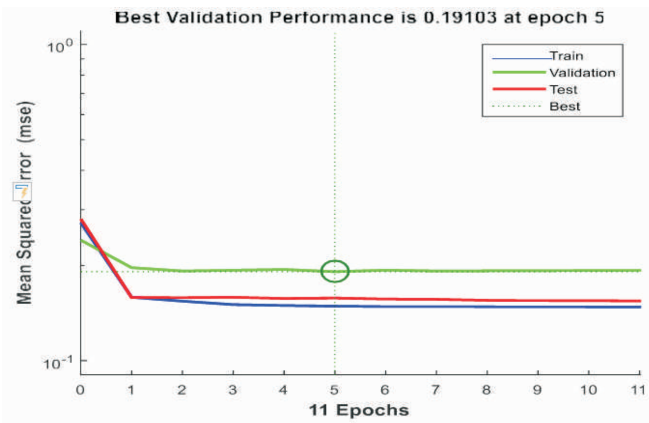

Figure 13. Mean Squared Error of the Model

Accuracy (%)	Correct Rate (%)	Error Rate (%)	Sensitivity (%)	Specificity (%)
55.1	69.1	30.9	55.2	55.4
50.1	54.5	45.5	55.2	53.7
55.4	61.9	38.1	60.2	62.2
77.897	78.4	21.6	78.11	78.2
65.3	62.1	37.9	64.3	62.5
55.113	60.12	39.88	58.9	60.3
65.4	66.5	33.5	67.78	68.87
69.97	64.99	35.01	66.76	68.13
65.55	67.87	32.13	60.67	63.33

Table 2. Performance Evaluation of Activation Function Combinations and Dynamic Thresholding of the Predictive Model

Conclusion

CAPTCHA has enhanced the security of web-based traffic in ensuring bot communication masquerading as human is mitigated. Research-based attempts in breaking

CAPTCHA have significant impacts in boosting the security and reliability of CAPTCHA technology. Artificial Intelligence has also been employed in this technology. Malicious users have made efforts to circumvent CAPTCHA protection in online services.

In particular, miscreants have devised means of adulterating CAPTCHA protection by smuggling fake CAPTCHA to intercept legitimate online services. This CAPTCHA smuggling attack creates a CAPTCHA challenge until the unsuspecting victims provide the required information. CAPTCHA smuggling attack is very lightweight in terms of required resources. Therefore, it is trivial for the bot master to add the required functionality to the existing bot program without limiting the existing functionality of the botnet. From the existing literature, there is no mechanism to detect CAPTCHA smuggling attack. The focus of the work is to develop a predictive model for detecting this attack. The outcome of this research will benefit web developers, web users, web hosting companies and internet service providers.

In order to achieve the goal of the work, hyperparametric framework was developed followed by multi-layered network connection using supervised deep learning approach. This serves as the basis for the development of the predictive model. The model was trained and validated using dynamic thresholding algorithm. Thereafter, it was tested and evaluated using a dataset sourced from W3School browser statistics containing usage statistics for both current browsers as well as several now defunct browsers as a result of smuggled attack. The associated results were presented and discussed. In particular, Accuracy of 77.897% was achieved at consistency of 0.1543 at threshold value of 0.55. The sensitivity and specificity of the model are 78.11% and 78.2%.

Acknowledgment

The researcher duly acknowledge the effort of research team in the Digital Forensic Laboratory of Federal University of Technology, Minna, Nigeria for making their facilities available for the experimentation of this work. Also worthy of mentioning is MBR Computer Consultants Ltd standby data house server for continuous iterations during the course of the model development.

References

[1]. **Alsuhibany, S. A. (2016).** A benchmark for designing usable and secure text-based captchas. *International Journal of Network Security & Its Applications,* 8(4), 41-54.

[2]. **Bilge, L., Strufe, T., Balzarotti, D., & Kirda, E. (2009, April).** All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th International Conference on World Wide Web* (pp. 551-560). ACM.

[3]. **Chen, J., Luo, X., Guo, Y., Zhang, Y., & Gong, D. (2017).** A Survey on Breaking Technique of Text-Based CAPTCHA. *Security and Communication Networks,* 2017, 1-15.

[4]. **Chew, M., & Tygar, J. D. (2004, September).** Image recognition CAPTCHAs. In *International Conference on Information Security* (pp. 268-279). Springer, Berlin, Heidelberg.

[5]. **Chilluru, M., Naick, B. R., & Nirupama, P. (2015).** Captcha based Password Authentication-A New Security Scheme. *International Journal of Computer Science and Information Technologies*, 6(4), 3514-3522.

[6]. **Chow, R., Golle, P., Jakobsson, M., Wang, L., & Wang, X. (2008, February).** Making captchas clickable. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications* (pp. 91-94). ACM.

[7]. **Egele, M., Bilge, L., Kirda, E., & Kruegel, C. (2010, March).** Captcha smuggling: hijacking web browsing sessions to create captcha farms. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 1865-1870). ACM.

[8]. **Gupta, S., & Garg, R. (2015).** Taxonomy of tools and techniques for network monitoring and quality assurance in 3G networks. *International Journal of Computer Applications,* 120 (21), 34-41.

[9]. **Hernández-Castro, C. J., R-Moreno, M. D., Barrero, D. F., & Gibson, S. (2017).** Using machine learning to identify common flaws in CAPTCHA design: FunCAPTCHA case analysis. *Computers & Security,* 70, 744-756.

[10]. **Nguyen, V. D., Chow, Y. W., & Susilo, W. (2014, May).** A

CAPTCHA scheme based on the identification of character locations. In *International Conference on Information Security Practice and Experience* (pp. 60-74). Springer, Cham.

[11]. Sharma, S., & Seth, N. (2015). Survey of Text CAPTCHA Techniques and Attacks. *International Journal of Engineering Trends and Technology (IJETT),* 22(6), 240-245.

[12]. Sivakorn, S., Polakis, I., & Keromytis, A. D. (2016a). I am robot: (deep) learning to break semantic image captchas. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on* (pp. 388-403). IEEE.

[13]. Sivakorn, S., Polakis, J., & Keromytis, A. D. (2016b). *I'm not a human: Breaking the Google reCAPTCHA. Black Hat*. Retrieved from https://www.blackhat.com/docs/asia-16/materials/asia-16-Sivakorn-Im-Not-a-Human-Breaking-the-Google-reCAPTCHA-wp.pdf

[14]. Uzun, E., Chung, S. P. H., Essa, I., & Lee, W. (2018). rtCaptcha: A real-time CAPTCHA based liveness detection system. In *Proceedings NDSS 2018: Network and Distributed System Security Symposium* (pp. 1-15).

[15]. Von Ahn, L., Blum, M., & Langford, J. (2004). Telling humans and computers apart automatically. *Communications of the ACM,* 47(2), 56-60.

[16]. Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003, May). CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 294-311). Springer, Berlin, Heidelberg.

ABOUT THE AUTHORS

Moses Olutayo Omoyele is currently a Post Graduate student in the Department of Cyber Security Science at the Federal University of Technology, Minna, Niger State. He holds a B.Sc. in Computer Science and a Masters degree in Computer Science from the University of Lagos. A former staff of College of Medicine University of Lagos, he along with his wife manages Techton Technologies and Services, a foremost ICT firm with branches in Abuja and Lagos.

Dr. Joseph A. Ojeniyi, is a Lecturer in the Department of Cyber Security Science, School of Information and Communication Technology, Federal University of Technology (FUT) Minna, Nigeria. He currently serves as the Chairman of Conference Organizing Committee of the faculty, 'ICTA 2018'. He received his PhD in Cyber Security Science from the same University, M.Sc. in Computer Science from University of Ibadan, Nigeria and a B.Tech in Mathematics/Computer Science from FUT Minna, Nigeria. He has been appointed as a reviewer to several indexed Journals. His area of interest includes Cyber Security, Digital Forensics, Deep Learning, Artificial Intelligence in Information Assurance/Security and Cyber Physical Systems.

Dr. Olawale S. Adebayo is a Lecturer in the Department of Cyber Security Science, Federal University of Technology Minna. He has a PhD from International Islamic University of Malaysia. He is a Member of several Professional bodies. In addition to his numerous responsibilities, he is the Research Coordinator of the Department. His research articles are published in reputable International Journals and Conferences are to his Credit.

i-manager®
Publications

3/343, Hill view, Town Railway Nager, Nagercoil
Kanyakumari Dist. Pin-629 001.
Tel: +91-4652-276675, 277675

e-mail: info@imanagerpublications.com
contact@imanagerpublications.com
www.imanagerpublications.com

i-manager Printing Press