

Phishing Emails Detection Models: A Comparative Study

Rian Sh. Al-Yozbakly ¹, Mafaz Alanezi ²

^{1,2}Department of Software, College of Computer Sciences and Mathematics, University of Mosul, Mosul, Iraq

¹rian.21csp85@student.uomosul.edu.iq, ²mafazmhalanezi@uomosul.edu.iq

Received: August 31, 2023

Revised: September 16, 2023

Accepted: September 17, 2023

Abstract

Today, phishing emails are the largest problem affecting internet services because they upset customers and cost businesses money. Methods that use the Natural Language Processing (NLP) principles also have many limitations and exhibit a flawed performance, especially regarding the non-English languages (such as the Arabic languages), given the lack of NLP for the Arabic language and the fact that this language has a rich vocabulary that delivers the same grammar and meaning. In this paper, viewed the previous models presented by other researchers, and also presented their RAPH model for the purpose of phishing detection. Where the model relies in its work on textual analysis of the content of e-mail messages and compares them with special datasets that include most of the commonly used words in electronic phishing. The results showed the effectiveness of the RAPH model, as it achieved a correct detection rate for phishing messages with a rate of (98.4%), while it achieved an error rate for legal messages with a rate of 7.5%.

Keywords: Phishing email Detection, Natural Language Processing, Python Libraries, RAPH, NLP Features.

1 Introduction

Phishing remains among the most harmful cybercrimes for both people and businesses, according to a threat assessment from the Australian Cyber Security Centre. Also, it is one of the worst cybercrimes, phishing even spreads other attacks (Hameed and Gamagedara , 2016)(Christian and MacLellan , 2018).

The criminal uses email or other communication tools like WhatsApp, Viber, or Facebook Messenger pretend to be a reputable business (like HSBC Bank) or person (perhaps operating in an authoritative capacity)(Burns, Johnson, and Caputo , 2019).

Cybercriminals frequently use phishing emails to distribute malicious links and files that might persuade recipients to perform certain activities or provide them with personal information (Burns, Johnson, and Caputo , 2019). This is a result of people and business using email for communication more often. Healthcare is one industry where phishing schemes or ransomware that results from phishing emails are particularly common (Chernyshev, Zeadally, and Baig , 2019).

In 2022, it was found that phishing attacks is the most targeted free email domains like: Google, Microsoft, Media, and others. As shown in “Fig. 1” Google's domain attacks were peaked to 72% in the second quarter of 2022, while Microsoft's domain attacks peaked to 21% in the third quarter of 2022(APWG , 2022a)(APWG , 2022b).

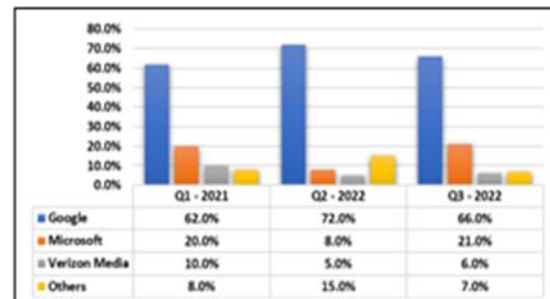


Figure 1. Targeted Free Webmail Providers

In this study, the proposed model was reviewed, and contributions were considered by creating a dataset of emails in the Arabic language and datasets of words and sentences commonly used in phishing. Also, the comparison with other models showed the showed the

researchers the accuracy of the proposed model in detecting phishing emails.

2 Related Works

This section addresses a set of models and methods used to detect phishing mail based on content.

Convolutional Keras word embedding and Neural Networks (CNNs) were utilised by (Hiransha et al., 2018) to detect phishing emails by focusing on the content. Two datasets—one with email headers and the other without—were compared by the authors. The findings demonstrated that, when email headers are ignored, the model gets better detection accuracy (96.8%).

(Peng, Harris, and Sawa , 2018) suggested a model (SEAHound) based on NLP techniques. This model processed a document, and examined email content one phrase at a time, also using two email datasets, the phishing emails used the dataset compiled by Jose Nazario 2005, and the legitimate emails used the Enron Corpus 2004 and achieved a 95% accuracy rate.

For the purpose of detecting phishing emails, (Fang et al. , 2019) presented the multilayer RCNN model and Themis model. Concurrently simulating email headers, character levels, bodies and word levels was done using Themis. They looked at four different datasets, including the Initial Security and Privacy Analysis Anti-Phishing Task (IWSPA-2018), Enron 2015, Nazario 2019, and artificial emails using Dada engine 2019, and they were able to classify the data with an outstanding accuracy of 99.848%.

(Maleki and Ghorbani , 2019) introduced for the purpose of discovering Business Email Compromises (BEC) content in emails a K-means-based Machine Learning (ML) algorithmic classification technique. Using Enron 2017 dataset, this approach has a 92% accuracy rate for classifying BEC assaults.

(Wei et al. , 2019) used the Word embedding to produce CNN layers with an accuracy of 86.43% and dense layers with an accuracy of 86.54%. The combined strategy resulted in an accuracy of 86.63%.

(Halgaš, Agrafiotis, and Nurse, 2020) proposed a novel automated system aiming to mitigate the threat of phishing emails with the use of Recurrent Neural Networks (RNNs). They used two types of datasets, SpamAssassin and Nazario and Enron and Nazario.

The findings imply that the system has a competitive advantage over the expert feature selection method, which is frequently used in Machine Learning-based efforts at phishing reduction.

(Y. Lee, Saxe, and Harang , 2020) introduced the Context-Aware Tiny Bert (CatBERT) model, which, at 1% false positive rates, outperforms the baselines of logistic regression and Distil Bidirectional Encoder Representations from Transformers (DistilBERT), Long Short-Term Memory (LSTM) with detection rates of 87%, 79%, and 54%, respectively. This model is more responsive to adversarial assaults that purposefully substitute keywords with typos or synonyms and is quicker than rival transformer techniques. CatBERT is 15% smaller and 160% faster than DistilBERT, the authors are employed by Sophos and used a big dataset (5 million emails) that Sophos had collected.

(Sonowal , 2020) suggested strategy which makes use of four groups of elements, including the email topic, body, readability and hyperlinks of contents. 41 features in all were chosen from the four dimensions. The gathered a dataset of legitimate emails from csmining group 2017, and phishing email from Jose Nazario's dataset 2017. The outcome demonstrates that the accuracy of the Binary Search Selection of Features (BSFS) approach was evaluated at 97.41% in comparison with Sequential Forward Feature Selection (SFFS) (95.63%) and Without Feature Selection (WFS) (95.56%).

D-Fence is a multimodule, comprehensive, and adaptable phishing email detection system that was created and presented by (J. Lee et al. , 2021). The structural module, text module, and URL module are the three separate analysis modules that provide D-Fence with the ability to cover a greater attack surface than competing products. They used two types of datasets, Enterprise email samples (EES) 2020, and collecting another dataset from Commercial cyber security company. D-Fence offers great detection capabilities with a high recall of 0.99 at a low false-positive rate of 1 in 10K, according to assessments on a real-world workplace email dataset.

For the purpose of identifying Business Email Compromises (BEC) and phishing material in emails, (Salahdine, Mrabet, and Kaabouch , 2021) suggested

using the Support Vector Machine (SVM), Linear Regression (LR), and Artificial Neural Network (ANN) algorithms. They used a dataset from real attacks launched against the email service of the University of North Dakota. In the suggested models' accuracy is 94.5%, 77.3%, and 92.9%, respectively.

For the purpose of identifying BEC and phishing material in emails, (Dutta, 2021) presented an approach using LSTM and RNN. They used dataset from Phishtank.org website and AlexaRank dataset. The suggested model has a 94.8% accuracy rate.

In order to identify BEC and phishing material in emails, (Ripa, Islam, and Arifuzzaman, 2021) indicated the using of the Random Forest (RF), SVM, LR, K-Nearest Neighbor (KNN), and Decision Tree (DT) algorithms. In RF, SVM, LR, KNN, and DT, respectively, the suggested models obtain accuracy of 96.8%, 96.6%, 92.28%, 94.09%, and 96.47%.

(Butt et al., 2022) suggested to identify BEC and phishing content in emails, Collected the dataset from CSDMC_SPAN online site. The created a feature extracted CSV and label files using the Naive Bayes (NB), LSTM, and SVM algorithms. The classification of phished emails is considered. Based on the analysis and implementation, the performance in recognizing email phishing attacks is better and more accurate. Email attacks were effectively classified with the highest degree of accuracy by the SVM, NB, and LSTM classifiers (99.62%, 97%, and 98%, accordingly).

(Bountakas and Xenakis, 2023) introduced the HELPHED phishing email detection technique, which combines ensemble learning techniques with hybrid characteristics to detect phishing emails. They used Enron 2004 dataset. By combining the linguistic and content characteristics of emails, the hybrid features accurately reflect emails. Another well-known Machine Learning/Deep Learning algorithms and current studies, obtaining an F1-score of 0.9942.

3 Proposed Methods

3.1 Datasets

Due to the limited availability of a dataset of phishing emails in the Arabic language, the first step was to collect a dataset from more than one dataset of legal and phishing emails in English, as the number of emails collected reached approximately 4,000 Emails,

distributed over 3250 legal emails and 750 phishing emails.

Work was carried out on this collected dataset in two directions. The first direction is to re-sort and manually check them, limit the emails based on the content of the emails, and extract them to obtain 1,000 legal Emails and 250 phishing emails. The second one was analyzing the content of messages, extracting words and phrases commonly used in phishing, and creating two datasets, the first for commonly used words in phishing emails, and the second dataset for phrases commonly used in phishing emails.

Concerning the obtained dataset, they were translated into Arabic using more than one translation program, where a Python libraries code was used to translate the content of emails, as well as using Google Translator. After comparing the translation using the two approaches, saving the two translations together with the original English text in one file. Finally, manually reformulating the sentences in each email so that the content's meaning is preserved, and the sentence context is maintained in line with Arabic grammar standards. The process of analysis and formation of datasets was divided into three sections:

1. Sentences dataset.
2. Words dataset.
3. Roots for the words dataset.

1. Sentences dataset:

A large group of phishing emails were analyzed and read, and the most used sentences in phishing were extracted and a special dataset was formed. "Fig. 2" shows a number of sentences in the dataset for sentences commonly used for phishing that consist of 420 sentences.

اجراءات الأمان	استعادة الحساب
اجراءات التحقق	استلام الطلب
اختراق الحساب	استلام مكافأة
ادارة الحساب	اعادة الاتصال
ارجاع الأموال	اعادة التأهيل
ارسال الطلب	اعادة تفعيل البطاقة الائتمانية
ارسال رمز التحقق	اعادة تفعيل الحساب
ارسال فواتير مستحقة	الحصول على الإصدار التجريبي
استرداد أموالك	الحصول على الجائزة

Figure 2: A number of Sentences Commonly Used for Phishing

2. Words Dataset:

Several phishing emails were examined for this type of analysis, commonly phishing words were extracted, and a dataset had more than 200 words that were often used in phishing messages. As seen in “Fig. 3” which shows a part of these words.

الاتصال	استلام	الرابط
اجراء	اشترك	الرصيد
احتمالية	اصدار	السجلات
اخبار	اضغط	السر
ادخال	اعادة	السفر
ارجاع	اغاني	السياسة
ارسل	اغلاق	السيبراني
ازالة	افتح	الشحن
استثمار	الاجازة	الشخصي
استرداد	الاجتماعي	الشراء
استعادة	الاحتيايل	الشركة
استفسار	الاحتيايلي	الشروط
استكشاف	الاختراق	الصفة

Figure 3: A Part of Words Commonly Used for Phishing.

3. Roots of the words dataset:

In this section, a dataset for the roots of common phishing words was not created; instead, it was done programmatically when running the model, where the roots of each word in the dataset of commonly used phishing words were extracted as well as the roots of the contents of the email's words from emails, and the comparison process was carried out. “Fig. 4” shows a number of words and their roots.

حجز	يحجز, الحجز
صدر	اصدار, يصدر
رسل	يرسل, ارسال
فعل	يفعل, تفعيل
أكد	يوكد, تأكيد
اتصل	اتصال, يتصل

Figure 4: A number of words and their roots.

3.2 Suggested Model (RAPH Model)

The model was built using the Python language and its libraries, and the tasks to be executed in the model were divided into two files. The first file calls several Python libraries, including the PySimpleGUI library, which creates windows. A small window appears, as in “Fig.

5”, which requests that enter the email information in order to contact it.

Figure 5. RAPH Model Entry Data Form

When the information is correct, starting execution of the second file, which first executes the codes for importing many libraries related to the Python language, such as the Imaplib, Email, Re, Pandas, NLTK, and Openpyxl libraries, using each library for a special purpose. “Table 1” shows the job for each library.

Table 1: The Tasks of Python Libraries in RAPH Model.

Library Name	Purpose of Library
PySimpleGUI	Creating the forms of the model.
Subprocess	Executing codes in python file from another python file.
Imaplib	Connecting to the email accounts by IMAP protocol.
Email	Analyse and read the email messages.
Re	Pattern matching and text manipulation.
Pandas	Reading the sentences from Excel file.
Arrow	Re-formatting the Date of emails.
NLTK	Analyse and processing the text.
Openpyxl	Reading and writing in Excel files.
DateTime	Importing the dates from emails.
Translator	Translate from English to Arabic language.
PyArabic	Processing the Arabic text.

The next step is loading and reading the datasets of words and sentences commonly used for phishing emails. After completing all the previous steps of connecting, loading the datasets, and importing the libraries, the system starts reading and processing e-

mails one by one. "Fig. 6" shows the flowchart of model implementation.

The system begins to check whether there is a tab folder named "Phishing" in the email account or not. If it is not available, the form will create a new tab folder and name its "Phishing", and then move on to the next step.

If the tab folder was previously created, it will go to the next step, which is to read the stored date from a special file to a special variable and store the current date to this file, then read the date of the first email

(which means the last email message received by the inbox).

Figure 6: Flowchart of RAPH Model

If the email date is older than the stored date, the execution will be stopped and exiting from the model, because this means the email did not receive any new emails and all old emails already did the scan of detection for them. When the email date is newer than the stored date, in this case, the model will go to read the contents of the email message.

Also, the content of the email message will be verified. If it is not in Arabic, the system will ignore the email message and go to read the next email. But if the content of the email message is in Arabic, it will process and delete some parameters that are not required in comparison operations, as shown in "Fig. 7", (such as numbers, special characters, and spaces). The numbers have been deleted because their presence does not suggest that the email is phishing because most of the legitimate emails contain mobile numbers as well as numbers for sums of money, so they were deleted because they do not help us in making comparisons.

The third one boils down to analyze the content of the email message into small sentences, comparing each sentence with the dataset of the sentences most used in phishing attacks, and storing the number of available iterations in a special variable symbolized by the letter (Z).

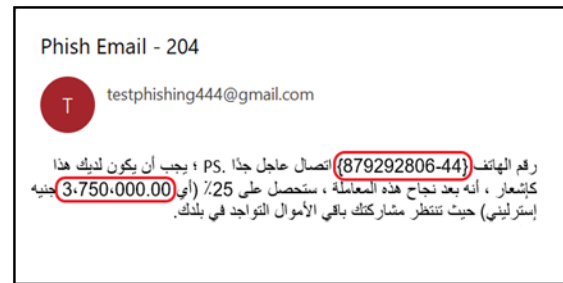
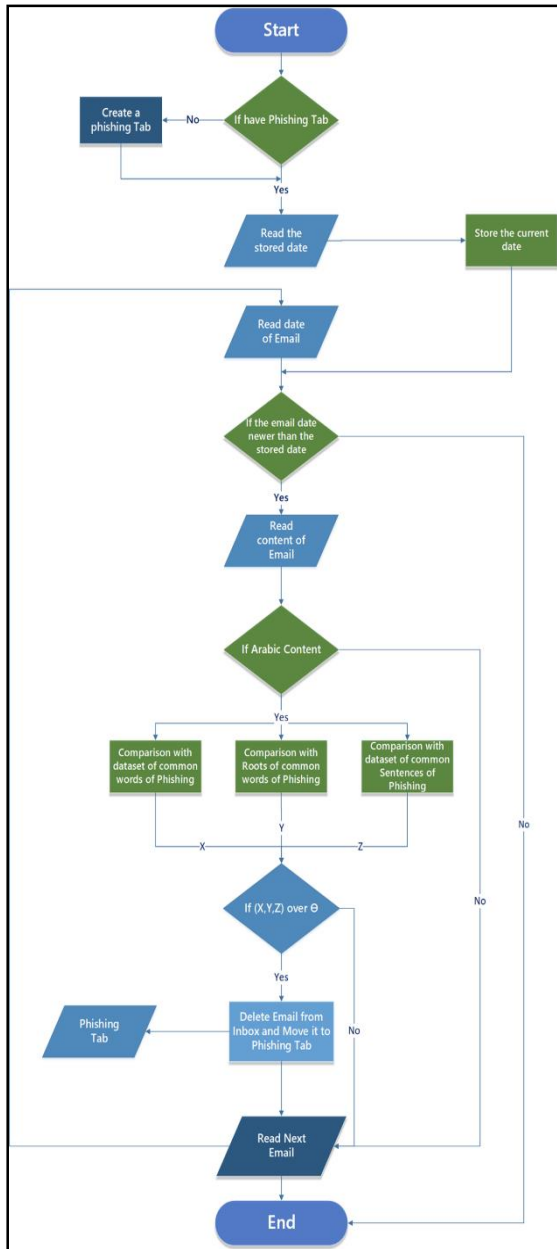


Figure 7: Deleting Some Parameters in the Email's Content

After completing the three comparisons above, the values of the three variables (X, Y and Z) are verified with the default values of the threshold in the model. If the values of the variables are equal to or greater than the default values, the model will decide that this email message is a phishing message and delete it from inbox and transferred to the tab folder named "Phishing", that was created at the beginning of the model execution.

But if the values of the variables are less than the default values of the model, it will consider that this

email message is a legitimate message and does not represent any danger in its presence in the inbox, and therefore it will bypass it and go to reading a new message and repeat all the above steps until it reaches the last email message in the inbox and then model execution is terminated.

To ensure that the model works perfectly and at an acceptable speed of execution, and that comparisons of previously existing emails in the inbox are not repeated at each execution of the model, the time factor of the email arrived at the inbox was relied upon.

Where a file was created in which the time of the last execution of the model was stored, and when the model run once again, this value (date) is called and the date of each email in the inbox is called, and compared if it is newer than the date of the last execution to be processed, and if it is older than the date of the last execution to be discarded. Thus, the implementation will be very fast, because it only checks new emails every time.

4 Experimental Results and Discussion

Each model or system contains default threshold values through which comparisons are made with the values extracted from the verification and processing operations carried out by the model on the selected datasets. The current study's model there are three methods for comparison, that is, there are three default threshold values for the model to make its right decision.

The model was executed several times by changing several threshold values until the values that gave us the best two results were reached. first result was (X=2, Y=1 and Z=1) and the second result was (X=1, Y=1 and Z=1).

Where the default value corresponds to the value of the variable X, which represents the number of similar words in the dataset of the most common words in phishing compared with the words contained in the content of the email.

While the value of the variable Y represents the number of similar word roots in the dataset of the most common words in phishing compared with the word roots contained in the content of the e-mail.

The value of the variable Z, which represents the number of similar sentences in the dataset of the most

common sentences in phishing compared with the sentences contained in the content of the email.

Because the study has three variables representing three comparisons, this means that there are 8 possibilities on the selected dataset. All possibilities have been implemented and tested. "Table 2" shows the results achieved in detecting phishing and legitimate emails.

To verify the results, we used the True Positive Rate (TPR) to detect the highest accuracy % for detecting phishing emails and used False Positive Rate (FPR) to detect the lower accuracy for detecting a legitimate email as a phishing email.

The results of the probabilities showed a striking variation: the probability (1, 3, 5, and 7) had the highest TPR for detecting phishing emails, while probabilities (8, 7, 6, and 5) had the lowest percentage of mistake for recognizing legitimate emails. The probability (5 and 7) produced the greatest results when determining the accuracy rate in identifying the two kinds of legitimate and phishing emails combined.

$$\text{Average of Accuracy} = (\text{TPR for Phishing emails} + \text{FPR for Legitimate emails}) / 2.$$

Table 2: Results of all Comparisons possibilities

Seq. No	Variables values			Phishing Detection		Legitimate Detection		Average of Accuracy
	X	Y	Z	TP (%)	FP (%)	TP (%)	FP (%)	
1	1	1	1	98.4	1.6	7.5	92.5	95.45 %
2	1	1	2	95.2	4.8	7.4	92.6	93.9 %
3	1	2	1	97.2	2.8	7.3	92.7	94.95 %
4	1	2	2	94.0	6.0	7.2	92.8	93.4
5	2	1	1	96.0	4.0	0.5	99.5	97.75
6	2	1	2	86.8	13.2	0.4	99.6	93.2
7	2	2	1	94.4	5.6	0.2	99.8	97.1
8	2	2	2	85.4	14.6	0.1	99.9	92.65

Authors and Model Name	Summary of Contribution	Methods	The used Dataset	Accuracy	Limitations
(Hiransha <i>et al.</i> , 2018)	In-depth instructions on how to spot phishing emails, including BEC attacks, and avoid falling for their traps are given in the paper.	CNN & Keras	---	96.8%	The dataset is not representative of real scenarios and a new real dataset is needed.
(Peng, Harris and Sawa, 2018) - SEAHound	This paper suggested a model (SEAHound) that analyzed a document and email's content sentence by sentence was proposed using NLP techniques.	NLP	Nazario 2005 Enron Corpus 2004	95.0%	The dataset applied does not take into consideration the demand for text-based emails rather than images. need a new dataset of text emails.
(Fang <i>et al.</i> , 2019b) - Themis	This research, a Themis model for phishing email detection was introduced. Concurrently simulating email headers, bodies, character levels, and word levels was done using Themis.	RCNN	IWSPA 2018 Enron 2015 Nazario 2019 Dada Engine 2019	99.848%	The dataset is not real phishing, especially which are generated by Dada Engine.
(Maleki and Ghorbani, 2019)	This study presents a machine learning algorithmic categorization strategy based on K-means that is designed to identify BEC content in emails.	K-means	Enron 2017	92.0%	There is no dynamic feature option available. Furthermore, the dataset chosen is not representative of actual data circumstances.
(Wei <i>et al.</i> , 2019)	The article produced CNN layers with an accuracy of 86.43% and dense layers with an accuracy of 86.54% using Word embedding. A total of combined two strategies were used, and the accuracy was 86.63%.	CNN	PhishTank 2019 Alexa 2019 Hphosts 2019 Joewein 2019 Malwaredomains 2019	86.63%	Not using and combining more techniques to achieve results with a higher accuracy rate.
(Halgaš, Agrafiotis and Nurse, 2020)	This study suggests an innovative automated approach that makes use of RNNs to lessen the danger posed by phishing emails.	RNN	SpamAssassin and Nazario Enron and Nazario	98.91% 96.74%	The dataset is not representative of real scenarios and a new real dataset is needed.
(Lee, Saxe and Harang, 2020) - CatBERT	In contrast to the baselines for DistilBERT, LSTM, and logistic regression, which are 83%, 79%, and 54%, respectively, the study developed a model called CatBERT, which has an 87% detection rate. Compared to DistilBERT, CatBERT is 15% smaller and 160% faster.	DistilBERT LSTM LR CatBERT	Sophos	83.0% 79.0% 54.0% 87.0%	Used the special dataset (Sophos company attacks), a new real dataset is needed.
(Sonowal, 2020b)	This study looks at four groups of elements, including email title, email body, hyperlinks, and accessibility of contents. 41 characteristics were ultimately chosen from the four categories. The outcome demonstrates that the (BSFS) is superior to the (SFFS) and the (WFS).	BSFS SFFS WFS	Nazario 2017 Csmining group 2017	97.41% 95.63% 95.56%	Create the greatest feature set, additional features must be added, and better feature selection techniques must be used.
(Lee <i>et al.</i> , 2021) - D-Fence	This study suggested a multimodule, thorough, and flexible phishing email detection technique named D-Fence. The three distinct analytic modules that give D-Fence the capacity to cover a larger attack surface than rival products are the structure module, text module, and URL module.	CNN LSTM	Enterprise email samples EES 2020 Commercial cyber security company	99%	Since the EES 2020 dataset is confidential, the experimental results from that dataset cannot be directly compared to those from subsequent investigations. The most effective and economical design demonstrates how the modularized system may dynamically lower its training and testing costs.
(Salahdine, Mrabet and Kaabouch, 2021)	The SVM, LR, and ANN algorithms were suggested in this paper as a way to identify BEC and phishing components in emails. using a dataset from attacks on the email system of the College of North Dakota.	SVM LR ANN	University of North Dakota	94.5% 77.3% 92.9%	The suggested models do not take notice of the email header and only employ the email content to identify phishing.
(Dutta, 2021)	This research proposed a long-short-term memory (LSTM) and recurrent neural network (RNN) approach. The model's accuracy rating is 94.8%.	RNN LSTM	Phishtank.org	94.8%	The accuracy reached by the recommended technique is still insufficient, and a more precise detection model is required.
(Ripa, Islam and Arifuzzaman, 2021)	The author proposed using machine learning to detect URLs. A recurrent neural network approach is used to detect phishing URLs. The research' findings show that the suggested strategy outperforms more recent methods in terms of recognizing dangerous URLs.	RF SVM KR KNN DT	---	96.8% 96.6% 92.28% 94.09% 96.47%	---
(Butt <i>et al.</i> , 2022)	The NB, LSTM, and SVM algorithms were utilized in this study to create a feature-extracted CSV file and tag file. The classification of phished emails is considered.	NB LLSTM SVM	CSDMC_SPAN Online	99.62% 97.0% 98.0%	The recommended models just evaluate the email content to spot phishing and ignore the email header.
(Bountakas and Xenakis, 2023) - HELPHED	This Paper provides an ensemble learning phishing email detection technology that uses Stacking and Soft Voting to efficiently process hybrid features.	ML & DL	Enron 2004	99.42%	---
RAPH Model	Our model used the NLP Techniques for detection the phishing emails, which using three comparison elements.	NLP	Creating a new dataset contains 1250 Emails	98.4 %	---

An overview of each related researcher's model is given in "Table 3", with information about the RAPH model being included in the last row. This table includes the names of the authors and the models, a

synopsis of the models' contributions, their techniques, the datasets they utilized, their accuracy, and finally their limitations.

Table 3: Comparison of Models

5 Conclusions

The risks of electronic crimes, especially phishing crimes, have increased dramatically as a result of the rapid growth in global electronic transactions and the nearly complete abolition of paper transactions. This risk and gap appeared particularly in the countries of the region Arabic due to the severe lack of research and methods for detecting phishing in Arabic, especially detection methods that rely on naive analysis. In the area of phishing detection. Upon analyzing several studies from earlier research, it became evident that neither the processing of natural languages nor phishing message processing in Arabic were addressed. Therefore, our suggested RAPH model produced excellent results and percentages, with the best rate for phishing detection being 98.4% and the best rate for recognizing real communications being 92.5%.

Acknowledgements

The authors would like to thank the College of Computer Science and Mathematics, University of Mosul for supporting this work.

References

- APWG (2022a) ‘APWG Phishing Trends Report 2nd Quarter 2022’, Anti-Phishing Working Group (APWG) [Preprint], (September). Available at: <http://www.apwg.org/>.
- APWG (2022b) ‘Phishing E-mail Reports and Phishing Site Trends 4 Brand-Domain Pairs Measurement 5 Brands & Legitimate Entities Hijacked by E-mail Phishing Attacks 6 Use of Domain Names for Phishing 7-9 Phishing and Identity Theft in Brazil 10-11 Most Targeted Industry’, APWG Phishing Activity Trends Report 1st Quarter 2022, 1(1), p. 13. Available at: https://docs.apwg.org/reports/apwg_trends_report_q1_2022.pdf.
- Bountakas, P. and Xenakis, C. (2023) ‘HELPHED: Hybrid Ensemble Learning PHishing Email Detection’, *Journal of Network and Computer Applications*, 210, p. 103545. Available at: <https://doi.org/https://doi.org/10.1016/j.jnca.2022.103545>.
- Burns, A.J., Johnson, M.E. and Caputo, D.D. (2019) ‘Spear phishing in a barrel: Insights from a targeted phishing campaign’, *Journal of Organizational Computing and Electronic Commerce*, 29(1), pp. 24–39. Available at: <https://doi.org/10.1080/10919392.2019.1552745>.
- Butt, U.A. et al. (2022) ‘Cloud-based email phishing attack using machine and deep learning algorithm’, *Complex and Intelligent Systems* [Preprint]. Available at: <https://doi.org/10.1007/s40747-022-00760-3>.
- Chandra, I. (2019) ‘Project_BTECH_20’.
- Chernyshev, M., Zeadally, S. and Baig, Z. (2019) ‘Healthcare Data Breaches: Implications for Digital Forensic Readiness’, *Journal of Medical Systems*, 43(1). Available at: <https://doi.org/10.1007/s10916-018-1123-2>.
- Christian, L. and MacLellan, S. (2018) ‘Governing Cyber Security in Canada, Australia and the United States’, *Center for International Governance Innovation*, p. 48. Available at: <https://www.cigionline.org/sites/default/files/documents/SERENE-RISCweb.pdf>.
- Dutta, A.K. (2021) ‘Detecting phishing websites using machine learning technique’, *PLoS ONE*, 16(10 October), pp. 1–17. Available at: <https://doi.org/10.1371/journal.pone.0258361>.
- Fang, Y. et al. (2019a) ‘Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism’, *IEEE Access*, 7, pp. 56329–56340. Available at: <https://doi.org/10.1109/ACCESS.2019.2913705>.
- Fang, Y. et al. (2019b) ‘Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism’, *IEEE Access*, 7, pp. 56329–56340. Available at: <https://doi.org/10.1109/ACCESS.2019.2913705>.
- Halgaš, L., Agrafiotis, I. and Nurse, J.R.C. (2020) ‘Catching the Phish: Detecting Phishing Attacks Using Recurrent Neural Networks (RNNs)’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11897 LNCS, pp. 219–233. Available at: https://doi.org/10.1007/978-3-030-39303-8_17.
- Hameed, A. (2022) User ticketing system with automatic resolution suggestions. BARCELONA.
- Hameed, M.A. and Gamagedara, N.A. (2016) ‘A model for the adoption process of information system security innovations in organisations: A theoretical perspective’, *Proceedings of the 27th Australasian Conference on Information Systems*, ACIS 2016, pp. 1–12.

OPEN ACCESS

<https://jmcer.org>





- Hiransha, M. et al. (2018) 'Deep learning based phishing E-mail detection CEN-Deepspam', CEUR Workshop Proceedings, 2124(Iwspa), pp. 16–20.
- Ingle, P., Kanade, H. and Lanke, A. (2016) 'Voice Based Email System for Blinds Introduction', 3(1), pp. 25–30.
- Journal, I. (2022) 'Design and Implementation of LIDAR System for Distance Measurements', *Interantional Journal of Scientific Research in Engineering and Management*, 06(10). Available at: <https://doi.org/10.55041/ijserm16626>.
- Lee, J. et al. (2021) 'D-Fence: A flexible, efficient, and comprehensive phishing email detection system', *Proceedings - 2021 IEEE European Symposium on Security and Privacy, Euro S and P 2021*, (September), pp. 578–597. Available at: <https://doi.org/10.1109/EuroSP51992.2021.00045>.
- Lee, Y., Saxe, J. and Harang, R. (2020) 'CATBERT: Context-Aware Tiny BERT for Detecting Social Engineering Emails'. Available at: <http://arxiv.org/abs/2010.03484>.
- Maleki, N. and Ghorbani, A.A. (2019) 'A Behavioral Based Detection Approach for Business Email Compromises'. Available at: <https://unbscholar.lib.unb.ca/islandora/object/unbscholar%3A10122>.
- Paul, R. and Mukhopadhyay, N. (2021) 'A Novel Python-based Voice Assistance System for reducing the Hardware Dependency of Modern Age Physical Servers', *International Research Journal of Engineering and Technology*, (May), pp. 1425–1431. Available at: www.irjet.net.
- Peng, T., Harris, I. and Sawa, Y. (2018) 'Detecting Phishing Attacks Using Natural Language Processing and Machine Learning', *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*, 2018-Janua, pp. 300–301. Available at: <https://doi.org/10.1109/ICSC.2018.00056>.
- Ripa, S.P., Islam, F. and Arifuzzaman, M. (2021) 'The Emergence Threat of Phishing Attack and The Detection Techniques Using Machine Learning Models', in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, pp. 1–6. Available at: <https://doi.org/10.1109/ACMI53878.2021.9528204>.
- Saabith, A.L.S., Vinothraj, T. and Fareez, M.M.M. (2021) 'A review on Python libraries and Ides for Data Science', *International Journal of Research in Engineering and Science*, 09(11), pp. 36–53. Available at: https://www.ijres.org/papers/Volume-9/Issue-11/Ser-2/G09113653.pdf%0Ahttps://www.researchgate.net/profile/Vinothraj-Thangarajah/publication/357898994_A_Review_on_Python_Libraries_and_IDEs_for_Data_Science/links/620249344d89183b338b49c2/A-Review-on-Python-
- Salahdine, F., Mrabet, Z. El and Kaabouch, N. (2021) '2021 IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2021', *2021 IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2021* [Preprint].
- Smith, C. (2017) 'Arrow Documentation'.
- Sonowal, G. (2020a) 'Phishing Email Detection Based on Binary Search Feature Selection', *SN Computer Science*, 1(4). Available at: <https://doi.org/10.1007/s42979-020-00194-z>.
- Sonowal, G. (2020b) 'Phishing Email Detection Based on Binary Search Feature Selection', *SN Computer Science*, 1(4), pp. 1–14. Available at: <https://doi.org/10.1007/s42979-020-00194-z>.
- Srinivas, P. et al. (2020) 'Raspberry Pi Based Personal Voice Assistant Using Python', *International Journal of Engineering Applied Sciences and Technology*, 04(11), pp. 105–108. Available at: <https://doi.org/10.33564/ijeast.2020.v04i11.020>.
- Thripuranthakam, L. et al. (2022) 'Stock Market Prediction Using Machine Learning and Twitter Sentiment Analysis: A Survey', *International Journal of Research in Engineering, Science and Management*, 5(4), pp. 144–149. Available at: <http://journals.resaim.com/ijresm/article/view/1968>.
- Verma, A. and Sharma, B. (2022) 'Dynamic E-Certificate Designing with Automatic Mailing System using Python and SQLite3', (October). Available at: <https://doi.org/10.13140/RG.2.2.17907.20000>.
- Wei, B. et al. (2019) 'A deep-learning-driven lightweight phishing detection sensor', *Sensors (Switzerland)*, 19(19), pp. 1–13. Available at: <https://doi.org/10.3390/s19194258>.
- Yin, T. and Henter, R. (2018) 'Translate Python Documentation'.
- Zerrouki, T. (2023) 'PyArabic : A Python package for Arabic text', *Journal of Open Source Software*, 8, pp. 10–15. Available at: <https://doi.org/10.21105/joss.04886>.

Biography



Rian Sh. Al-Yozbaky    is obtained his Bachelor of Science (BSc) in Computer Sciences in 2005 from the Computer Sciences Department, Al-Hadba'a University College in Mosul, Iraq. Then he was appointed as an assistant programmer in the Governorate of Ninawa in 2012, After that, he got "Assistant Chief Programmer" in 2022. Currently, he is a master's student in Computer Science department, computer sciences and mathematics college, university of Mosul, Iraq. The subjects for interest, Cyber Security, Database management, Network Security, Designing, encryption and decryption. Email: rian.21csp85@student.uomosul.edu.iq



Mafaz Alanezi     is a faculty member at the Department of Computer Science, University of Mosul, Iraq. She obtained her Ph.D. degree in Computer Science in the field of Computer and Network Security from University of Mosul / Iraq in 2012. Her M.Sc. degree was also in Computer Science in the field of Image Processing from the University of Mosul/ Iraq in 2003. Her current scientific degree Prof. Dr in Cybersecurity and Information Security.

OPEN ACCESS

<https://jmcer.org>