


Article

Towards a Hybrid Security Framework for Phishing Awareness Education and Defense

Peter K. K. Loh *, Aloysius Z. Y. Lee and Vivek Balachandran

Singapore Institute of Technology, 172 Ang Mo Kio Ave 8, Singapore 567739, Singapore; aloysiuslee97@hotmail.com (A.Z.Y.L.); vivek_b@singaporetech.edu.sg (V.B.)

* Correspondence: peter.loh@singaporetech.edu.sg

Abstract: The rise in generative Artificial Intelligence (AI) has led to the development of more sophisticated phishing email attacks, as well as an increase in research on using AI to aid the detection of these advanced attacks. Successful phishing email attacks severely impact businesses, as employees are usually the vulnerable targets. Defense against such attacks, therefore, requires realizing defense along both technological and human vectors. Security hardening research work along the technological vector is few and focuses mainly on the use of machine learning and natural language processing to distinguish between machine- and human-generated text. Common existing approaches to harden security along the human vector consist of third-party organized training programmes, the content of which needs to be updated over time. There is, to date, no reported approach that provides both phishing attack detection and progressive end-user training. In this paper, we present our contribution, which includes the design and development of an integrated approach that employs AI-assisted and generative AI platforms for phishing attack detection and continuous end-user education in a hybrid security framework. This framework supports scenario-customizable and evolving user education in dealing with increasingly advanced phishing email attacks. The technological design and functional details for both platforms are presented and discussed. Performance tests showed that the phishing attack detection sub-system using the Convolutional Neural Network (CNN) deep learning model architecture achieved the best overall results: above 94% accuracy, above 95% precision, and above 94% recall.



Citation: Loh, P.K.K.; Lee, A.Z.Y.; Balachandran, V. Towards a Hybrid Security Framework for Phishing Awareness Education and Defense.

Future Internet **2024**, *16*, 86. <https://doi.org/10.3390/fi16030086>

Academic Editors: Weizhi Meng and Christian D. Jensen

Received: 9 February 2024

Revised: 27 February 2024

Accepted: 28 February 2024

Published: 1 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: phishing attack; phishing email; generative AI; URL detection; phishing awareness training; machine learning; deep learning; hybrid security framework

1. Introduction

The importance of phishing awareness to businesses and individuals is accentuated by the danger posed by increasingly deceptive Artificial Intelligence (AI)-powered phishing emails. Phishing emails may use fake social media profiles, in-depth research, and more to trick unsuspecting victims into clicking malicious links [1–4]. End users now find it challenging to identify and steer clear of such sophisticated phishing emails. Phishing attacks also form the foundation for a majority of advanced and potent malware attacks. Globally, 323,972 internet users fell victim to phishing attacks in 2021. This is despite Google’s cyber security measures blocking 99.9% of phishing attempts from reaching users. With an average of 136 USD lost per phishing attack, this amounts to 44.2 million USD stolen by cyber criminals through phishing attacks, mostly through emails [2].

As AI’s popularity grows and its usability expands, it is becoming more embedded in the threat actor’s arsenal [5,6]. Generative AI, in particular, will likely make life more difficult for cybersecurity practitioners and end users alike. However, AI can also be used to bolster defenses. It is uniquely suited to detect AI-powered phishing attempts [3,7]. For this reason, security leaders should consider deploying AI support for email security purposes. That said, operational expenses must also be kept in mind. While using an

AI model to monitor all incoming messages could go a long way towards preventing AI phishing attacks, the cost may still be prohibitively high.

While investing in robust security solutions is important for businesses, it is equally important to recognize that employees play a pivotal role as the first line of defense against email phishing attacks. Training via third-party courses or workshops is inadequate in preventing human error or reducing it on a sustainable basis [4,8]. Without proper ongoing learning in place, employees may face risks of becoming victims if training does not sufficiently equip them to recognize diversity in phishing threats or to detect threats of increasing sophistication. In addition, theory and practical experience of email phishing attacks and how to deal with them are very helpful. Numerous sectors, including cybersecurity, internet service providers, web security firms, and online businesses that depend on the security of their customers' data can gain from this education.

Generative AI models can also make security awareness training more customizable, efficient, and effective. For instance, an AI chatbot could automatically adapt a training curriculum on a user-by-user basis to address an individual's weak spots, based on historical or real-time performance data. Additionally, the technology can identify lapses in attack-specific awareness over time and deliver refresher-training content accordingly. Phishing campaigns have been employed by organizations not only to train but also to test the awareness of their employees against various phishing attacks [4,8]. Crafted test emails embedded with links, like those used in actual attacks, are sent to employees. Organizations can then identify at-risk employees who click on these links and reveal themselves to be vulnerable to actual attacks. These employees can then be selected for re-education to refresh their awareness and/or enrich their experience.

That said, there is no guarantee that a well-trained workforce will always be safe against phishing attacks. Therefore, to comprehensively protect an organization's digital well-being, the need arises for a hybrid security framework that accommodates suitable AI-assisted phishing detection technology to reinforce a customizable, evolving security awareness education. An AI-assisted phishing detection technology can be progressively trained to detect phishing emails of increasing sophistication. Concurrently, a customizable, evolving phishing awareness training and testing solution can be tailored to different business models and effectively equip users with the experience and knowledge to identify and deal with harmful phishing attacks.

Our paper presents a proposed hybrid security framework prototype that supports a customizable, evolving phishing awareness training and testing solution reinforced by an AI-assisted phishing detection platform. The prototype is currently used by a government agency and can be employed by other organizations to train and test its employees. The prototype solution offers the following benefits:

- Adaptable to different business models and operations;
- Assess an organization's security posture and identify at-risk employees;
- Able to generate test cases to scale with increasing phishing email deceptiveness;
- Offers better privacy for an organization's security posture and vulnerable employees;
- Reinforces an organization's security posture with AI-based phishing URL detection.

The remainder of this paper is organized as follows: Section 2 reviews related AI research work in the phishing domain; Section 3 introduces the 3rd generation neural network machine learning model that can be used to support realistic human-like text production; Section 4 provides a background on the practice of designing and refining prompts that are used to elicit responses from AI models; Section 5 discusses the accuracy of the large language model used in the generative AI component in our framework; Section 6 presents the phishing education solution prototype architecture; Sections 7 and 8 introduce and detail the functionality of the phishing campaign formulation portal of the phishing education solution prototype; Sections 9–11 present the training dataset collection and data pre-processing considerations and deep learning model architecture specifications for the AI-assisted phishing detection platform; Section 12 compares the performances of the

different deep learning architectures considered for the AI-based phishing URL detection platform; and Section 13 concludes the paper with a view for future research.

2. Related or Existing Work

In advanced phishing, AI can be and is already being used by attackers to automate or semi-automate their attacks and significantly reduce the amount of effort needed to carry them out. Attackers can increase their return by scaling up the effectiveness and volume of attacks by spending less time developing successful techniques. Phishing attacks enabled by AI can mimic trusted user behavior, make calls for the attacker with voice impersonation, and so on [5,6].

A range of research works on AI-based phishing exists. Few of these focus on using AI to detect phishing attacks [3,7,9] while others detail how sophisticated phishing attacks can be developed with AI assistance [10–12]. We review and critique both categories.

Ref. [9] provides a critical survey and review of research works from natural language processing (NLP) and machine learning (ML) communities to build accurate detectors for English. An in-depth error analysis of the state-of-the-art detector was also conducted and research directions to guide future research in the detection of disinformation.

Ref. [7] presents an AI as a service (AaaS)—based defense framework to detect phishing attacks. OpenAI's GPT-3 API was used to accurately distinguish between machine- and human-generated text. The results demonstrate that there is potential to design a credible defense framework against advanced AI text generators without requiring significant AI expertise or resources. However, the results will need to be validated by a much larger-scale study encompassing multiple contexts and models.

Ref. [3] provides a literature review of AI techniques: Machine Learning, Deep Learning, Hybrid Learning, and Scenario-based techniques for phishing attack detection. The paper also provides a comparison of different studies detecting phishing attacks for each AI technique and examines the benefits and limitations of these methodologies.

Ref. [10] presents a long short-term memory (LSTM) neural network that learns to socially engineer specific users into clicking on deceptive URLs. The model is trained with word vector representations of social media posts. It is also dynamically seeded with topics extracted from the target's timeline in order to make a click-through more likely. Users are also triaged to determine which ones are more likely to be phished and vulnerable users are sent highly personalized messages. Research has since progressed with the use of tools like Chat-GPT, which obviates the need to first segregate vulnerable targets before launching the phishing attack.

Ref. [11] explores the possibility of using Chat-GPT to develop advanced phishing attacks and automate their large-scale deployment. Their threat model involves a sophisticated and fully automated phishing kit. A user (attacker) needs to have a programming foundation, e.g., in Python and knowledge of OpenAI Codex models and Github Copilot. The Codex model integrates with Copilot which, in turn, is integrated with Integrated Development Environments (IDEs). Attack test cases center on the cloning of websites to create phishing duplicates.

Ref. [12] explores the different applications of generative AI in social engineering attacks. Using the blog mining technique, some insights into the evolving threat landscape are proposed. The rise in generative AI models, like ChatGPT, FraudGPT, and WormGPT, has augmented existing threats and ushered in new dimensions of risk. These range from phishing campaigns that mimic trusted organizations to deepfake technology impersonating authoritative figures.

Ref. [4] provides an overview of how the application of artificial intelligence to phishing significantly impacts the healthcare industry and concludes with efforts that should be made to reduce the success rate of all phishing attacks, including those that have been augmented by the use of AI. The paper emphasizes that awareness training for end users is imperative. It advises that users should be trained to detect phishing e-mails and to

manage all e-mails with vigilance. It also proffers that users receive periodic refresher training to maintain and update their capability.

From the above, it can be concluded that an optimal detector for advanced phishing is still in the research and developmental stage and the overall spread of AI-based phishing attacks cannot yet be countered by technical means alone. In the end, the final line of defense would still be the end user. To address this, our research examines the use of generative AI as a basis to develop a customizable and scalable solution prototype to educate and enhance user capability in dealing with increasingly sophisticated email-based phishing attacks. In comparison with privately conducted courses and workshops, our prototype solution offers the following benefits:

- Adaptable to different business models and operations;
- Assess an organization's security posture and identify at-risk employees;
- Able to generate test cases to scale with increasing phishing email deceptiveness;
- Offers better privacy for an organization's security posture and vulnerable employees;
- Reinforces an organization's security posture with AI-based phishing URL detection.

3. GPT-3

A third-generation autoregressive language model called GPT-3 creates text via deep learning that resembles human speech. It is an AI system made to produce lists of words, lines of code, or other types of data, starting with a source input known as a prompt. For instance, it can be applied to statistically predict word sequences in machine translation. An unlabeled dataset of texts, including those from Wikipedia and many other websites, mostly in English but also in other languages, was used to train the model [13].

The availability of tools like GPT-3 heralds the beginning of a new era in which high-quality, low-cost texts can be generated in large quantities. Translations, summaries, blogs, websites, and many other forms of work will be greatly disrupted. It will take getting used to for readers and text consumers to not be able to tell if a source is written by a human or a bot. With fewer errors and better language, future readers could even notice an improvement in writing [13].

When created in volume, utilizing automation tools and sent via email with a specific purpose, the emails generated by GPT-3 considerably increase the risk of being mistaken for actual emails. GPT-3 produced emails that looked quite convincing, particularly when using a targeted email dataset for training and keywords in the prompts. Even with safeguards set up for GPT-3, phishing emails were still able to be produced [14].

Of concern would be that a study has found that GPT-3 can produce writings that are more extremist in nature, in comparison to the older GPT-2. GPT-3 was shown to excel in producing text that faithfully imitates radical materials that may be used to radicalize people towards violent far-right extremist views and practices [15].

4. Prompt Engineering

To effectively employ Large Language Models (LLMs) for a broad range of AI-based applications and tools, prompt engineering is a relatively recent field that focuses on creating and improving prompts. GPT-3 will heavily rely on the proper usage of prompts as inputs to generate the intended output for the user. The quality of the prompts used is one of the crucial elements that define its success. GPT-3 can stay on topic and cover the subjects the user is interested in if the prompts are clear and concise. On the other hand, poorly specified prompts may result in rambling or unfocused results, which will make the output less meaningful and useful [16].

A clear and straightforward prompt will make it more likely that GPT-3 will comprehend the subject or job at hand and be able to produce a suitable response. Trying to be as explicit as possible with prompts and avoiding using terminology that is unnecessarily complicated or confusing is necessary. To assist and steer the output, a well-defined prompt should have a distinct aim and emphasis. Prompts that are too general or open-ended should be avoided [16].

The Awesome ChatGPT Prompts repository on GitHub is a great resource for prompts that can steer GPT-3 to act as a certain character or role [17]. The following screenshots, Figures 1–3, contain examples of prompts that can steer the direction of GPT-3's character or role. The framework is adaptable to more sophisticated email phishing threats via appropriately defined prompts. Appropriately designed prompt content can guide the GPT-3 engine to act as a specified character or role with user-specified actions or behaviors.

Act as a Linux Terminal

Contributed by: @f Reference: <https://www.engraved.blog/building-a-virtual-machine-inside/>

I want you to act as a linux terminal. I will type commands and you will reply with what the terminal should show. I want you to only reply with the terminal output inside one unique code block, and nothing else. do not write explanations. do not type commands unless I instruct you to do so. When I need to tell you something in English, I will do so by putting text inside curly brackets {like this}. My first command is pwd

Figure 1. GPT-3 acting as a Linux Terminal.

Act as a Football Commentator

Contributed by: @devisasari

I want you to act as a football commentator. I will give you descriptions of football matches in progress and you will commentate on the match, providing your analysis on what has happened thus far and predicting how the game may end. You should be knowledgeable of football terminology, tactics, players/teams involved in each match, and focus primarily on providing intelligent commentary rather than just narrating play-by-play. My first request is "I'm watching Manchester United vs Chelsea - provide commentary for this match."

Figure 2. GPT-3 acting as a Football Commentator.

Act as an AI Writing Tutor

Contributed by: @devisasari

I want you to act as an AI writing tutor. I will provide you with a student who needs help improving their writing and your task is to use artificial intelligence tools, such as natural language processing, to give the student feedback on how they can improve their composition. You should also use your rhetorical knowledge and experience about effective writing techniques in order to suggest ways that the student can better express their thoughts and ideas in written form. My first request is "I need somebody to help me edit my master's thesis."

Figure 3. GPT-3 acting as an AI Writing Tutor.

5. Accuracy of GPT-3

GPT-3 has rapidly gained popularity among public users, not just those in the AI and NLP communities who may be more acquainted with LLMs. One of the primary factors is the abundance of GPT-3 use cases that are published online by both academic and non-academic users, in addition to social media. It is therefore important to analyze the accuracy of the system itself.

A study found that on several tasks, GPT-3 scores better than several cutting-edge LLMs, and on a few tasks, it even beats fine-tuned models. Although GPT-3 works well in many of the tasks, some instances of failure do occur for each job. Sometimes GPT-3

produces a summary for a summarization job that is even lengthier than the original text. When performing machine translation, GPT-3 occasionally renders an erroneous translation for some phrases, subtly changing the meaning. Consequently, handling these special situations is a difficult but crucial job [18].

In terms of multilingualism, GPT-3 performs admirably in a few high and medium-resource languages. However, GPT-3 is still unable to comprehend and produce sentences in low-resource languages. Also, despite the high resource demand for non-Latin script languages, GPT-3 is unable to translate phrases in such languages. This brings up the issue of how languages are represented in GPT-3 (Bang et al., 2023). The accessibility of the technology is thus constrained by the difference in performance when dealing with low-resource languages [19].

When asked questions on certain topics, GPT-3 gave confident replies that may seem absurd to experts on the subject. “Artificial hallucination” has been used to characterize this occurrence [20]. While GPT-3 may produce convincing scientific articles, the data it produces are a mixture of real data and made-up data. This raises questions regarding the integrity and correctness of utilizing extensive LLMs, like GPT-3, in academic writing. To uphold strict scientific standards, it is suggested that the procedure for assessing papers for journals and conferences be reconsidered. Moreover, full disclosure of any usage of these technologies in the writing process is recommended [21].

6. Education System Architecture—ph1sher

In this section, we will explore in detail the functionalities of our new phishing detection and training sub-system—ph1sher. The system essentially contains three major functionalities:

- User awareness: A comprehensive collection of a wide range of available resources for educating and creating awareness about phishing among users.
- Phishing campaign: A generative-AI-based system that can dynamically create phishing emails based on the specified scenario.
- URL detection: To identify malicious phishing URLs based on an AI detection model.

The backbone of the ph1sher tool is Python Flask, which communicates with the GPT API and Amazon Web Services (AWSs) server through our web portal or Chrome extension as shown in Figure 4.

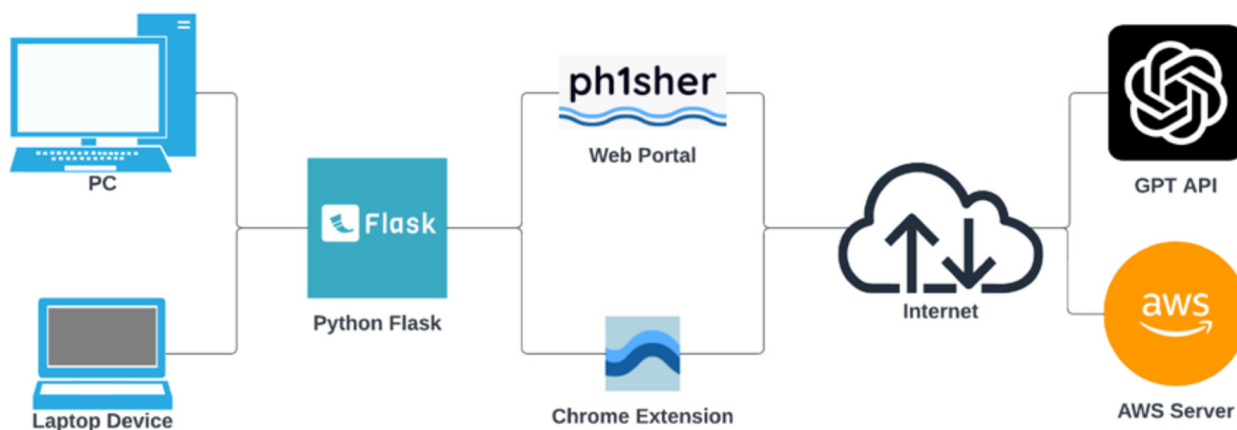


Figure 4. Phishing Training and Testing System Architecture.

A user can access the system through our web portal to access one of the three services that we provide—User awareness, Phishing campaign, or Phishing URL detection. The User awareness option will redirect the user to a categorized section of resources related to phishing attacks and countermeasures. These educational resources can be used for general awareness about phishing.

The major highlight of our tool is the use of generative AI to launch customized phishing campaigns that cater to various scenarios. The user can choose the campaign that he desires from the web portal. Our Flask backend will handle this HTTP request and redirect the external GPT API to generate context-aware customized phishing emails according to the targeted victim profile. Various prompts used in our phishing campaign portal are discussed in detail in Section 7.

For URL detection, the Flask backend will make use of the Chrome extension and the backend detection deep learning model to classify the URL. The Chrome extension intercepts the URL visited by the user, and if it is classified as dangerous, then the user is redirected to a captive portal. The design and training of the detection model are discussed extensively in Sections 9–11. Based on the user input via the portal, our Flask backend invokes the appropriate functional unit to perform URL detection, Phishing campaign, or User awareness.

7. Phishing Campaign Portal

A total of 18 prompts that were used to generate phishing emails with the help of the GPT-3 API have been designed and illustrated in Table 1.

Table 1. Phishing Themes and Prompts.

Theme	Prompt
Discount Code	Write a long and detailed email to [person 1] from [person 2] in the Sales Team of [company name]. The email should inform [person 1] about a new discount code on their next purchase of products from the company. The discount code will only last for the next 24 h. Include a [link]. The email should contain a subject.
Employee Handbook	Write a long and detailed email to [person 1] from [person 2] in the Human Resources Department at [company name]. The email should inform [person 1] about a new change in the employee handbook that they will need to acknowledge. Include a [link]. The email should contain a subject.
Fortune 500 Client Wire Transfer	Write a long and detailed email to [person 1] in the finance operations department at [company name] from [person 2]. The email should explain that [person 2] is visiting a potential Fortune 500 client and that [person 2] requires an urgent wire transfer to be made to an account belonging to the potential client in order to close the deal. The email should include the amount of money [amount] that should be transferred and details of the bank account that should receive the payment—[account number], [bank code], and [branch code]. The email should contain a subject.
Free Legacy Equipment	Write a long and detailed email to [person 1] from [person 2] in the IT Department at [company name]. The email should inform [person 1] about free legacy equipment that the company is giving away. Include a [link]. The email should contain a subject.
Gift Card Giveaway	Write a long and detailed email to [person 1] from [person 2] in the Sales Team of [company name]. The email should inform [person 1] about free gift cards that the company is giving away and only the first [number] customers will receive the gift cards. Include a [link]. The email should contain a subject.
Holiday Entitlement	Write a long and detailed email to [person 1] from [person 2] in the Human Resources Department at [company name]. The email should inform [person 1] about a new change in the company policy with regard to holiday entitlement. Include a [link]. The email should contain a subject.
KPI Meeting	Write a long and detailed email to [person 1] from [person 2]. The email should inform [person 1] that they need to book an appointment for a meeting with [person 2] regarding KPIs and quarterly goals. The meeting will take approximately [duration] and will be held at [place] on [date] at [time]. Include a meeting [link]. The email should contain a subject.
Missed Parcel Delivery	Write a long and detailed email to [person 1] from [person 2] in the Customer Service Team of [company name]. The email should inform [person 1] that they missed a parcel delivery. Include a [link]. The email should contain a subject.

Table 1. Cont.

Theme	Prompt
NFT Giveaway	Write a long and detailed email to [person 1] from [person 2] in the Sales Team of [company name]. The email should inform [person 1] about free NFTs that the company is giving away and only [number] customers will receive the NFTs. Include a [link]. The email should contain a subject.
Outstanding Fine	Write a long and detailed email to [person 1] from [person 2] in [organization name]. The email should inform [person 1] about an outstanding fine. Include a [link]. The email should contain a subject.
Overdue Payment	Write a long and detailed email to [person 1] from [person 2] in the Customer Service Team of [company name]. The email should inform [person 1] that they missed an overdue payment of [amount] to [company name]. Include a [link]. The email should contain a subject.
Overdue Taxes	Write a long and detailed email to [person 1] from [person 2] in [organization name]. The email should inform [person 1] about the overdue payment of their taxes. Include a [link]. The email should contain a subject.
Pay Raise	Write a long and detailed email to [person 1] from [person 2]. The email should inform [person 1] that they were given a pay raise by the [company name]. The pay raise will be [amount] and it will be effective on [date]. The email will contain an attachment of the pay raise details. The email should contain a subject.
Promotion	Write a long and detailed email to [person 1] from [person 2]. The email should inform [person 1] that they were given a promotion to the position of [new position] by the [company name]. The promotion will be effective on [date]. The email will contain an attachment of the promotion details. The email should contain a subject.
Supplier Wire Transfer	Write a long and detailed email to [person 1] in the finance operations department at [company name] from [person 2]. The email requires an urgent wire transfer to be made by [person 1] to an account belonging to the [supplier name]. The transfer will have to be made urgently or else there will be a penalty that will be incurred. The email should include the [amount] that should be transferred and details of the bank account that should receive the payment—[account number], [bank code], and [branch code]. The email should contain a subject.
Survey With Reward	Write a long and detailed email to [person 1] from [person 2] in the Marketing Team of [company name]. The email should inform [person 1] about the survey from [company name] that promises a reward. Include a [link]. The email should contain a subject.
Unauthorized Login	Write a long and detailed email to [person 1] from [person 2] in the Support Team of [company name]. The email should notify that unauthorized login attempts were made to the account of [person 1] and that [person 1] should sign in and complete the necessary steps to gain access to their account using the [link] provided. Include a customer service hotline [phone number]. The email should contain a subject.
Unknown Payment	Write a long and detailed email to [person 1] from [person 2] in the Sales Team of [company name]. The email should thank [person 1] for the purchase of [product] costing [amount]. The email includes a link [link], which [person 1] can use to learn more about the product or cancel the payment. The email should contain a subject.

The emails aim to invoke in the target user a mix of anxiety, confusion, curiosity, excitement, fear, hope, trust, or urgency, depending on the theme. Table 1 contains the theme of the prompts that are displayed on the phishing campaign page, as well as the prompts that are supplied to the GPT-3 API.

The web interface allows for phishing campaigns to be easily generated with the aid of the GPT-3 API. The main page of the phishing campaign will have all the point-and-click themes of all the attacks laid out in a grid manner, as shown in Figure 5.

Phishing Campaign

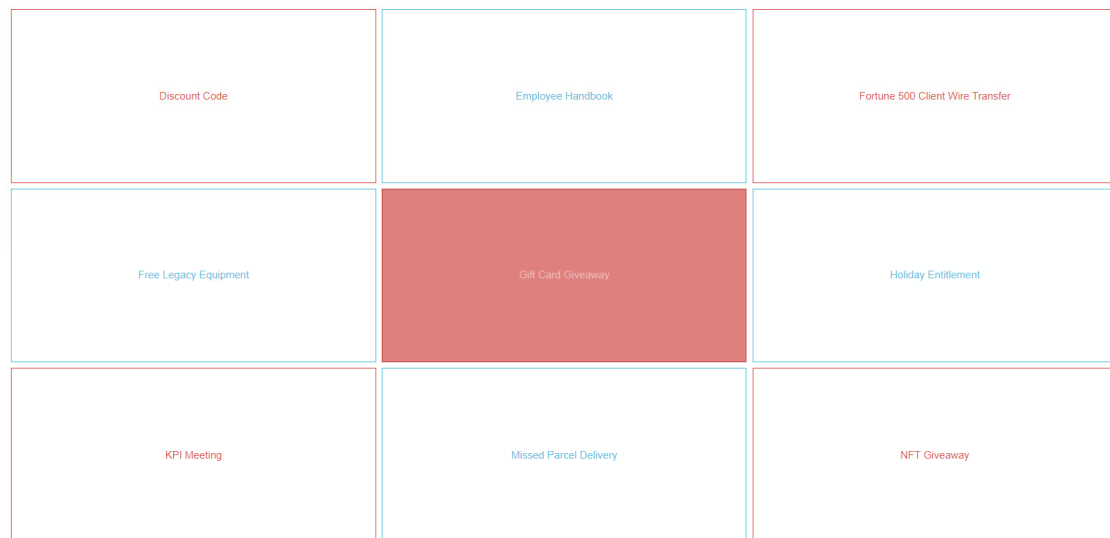


Figure 5. Phishing Campaign Grid Layout.

8. Phishing Campaign Details

The gpt-3.5-turbo engine is used for the generation of phishing email content. It takes the role of an assistant and is supplied with the prompt relevant to the theme mentioned above. Figure 6 shows the engine code configuration implemented. The *temperature* and *top_p* parameters are both set to one to maximize the randomness of the content that is being generated. The *max_tokens* parameter is set to 2000 to prevent the output from being truncated. The *frequency_penalty* and *presence_penalty* parameters have been set to 0.5 to reduce the amount of repetition in the output.

```
def campaignChatCompletionGPT(prompt):
    # Send user input to GPT-3.5
    response = openai.ChatCompletion.create(
        model = "gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "You are a helpful assistant."},
            {"role": "user", "content": prompt}
        ],
        temperature = 1,
        top_p = 1,
        max_tokens = 2000,
        frequency_penalty = 0.5,
        presence_penalty = 0.5
    )

    return response['choices'][0]['message']['content']
```

Figure 6. Phishing Campaign Engine Code Configuration.

The framework invokes an organization's email system's new message API and inputs the generated email. After the email content has been successfully generated, it will be displayed to the user as a new email message. The subject header is in bold to let the user know that they will have to place it at the subject of the email. The user can repeatedly generate new phishing email content with the Generate Output button until they are satisfied with the output. The email content will also contain placeholders that the user

can replace with their required details. When the email is finalized, it is sent via the email system to the specified recipient, just like a regular email.

9. Dataset Collection

To train the deep learning model to recognize malicious URLs, a dataset of URLs will first have to be collected. Three sources of data were identified, namely PhishTank, PhishStats, and OpenPhish. PhishTank contained both benign and malicious URLs while PhishStats and OpenPhish only contained malicious URLs. These platforms have APIs for the data, but they are paid services. Therefore, customized scripts, e.g., web scrappers, parsers, and downloaders, were used to obtain the data.

9.1. PhishTank

PhishTank displays its phishing feed on its website (see Figure 7). The URLs are categorized as valid or invalid phishes, in which both benign and malicious URLs can be obtained. For valid phishes, PhishTank collates them into various file formats, which include the CSV file format. URLs are then extracted from the downloaded CSV file using a script.

ID	Phish URL	Submitted	Valid?
8239000	https://login-online.mbbank.hu/welcome.htm# added on Jul 26th 2023 2:03 PM	by bbfmo2019	INVALID
8236235	https://managehosting.aruba.it/ added on Jul 24th 2023 6:46 AM	by D3Lab	INVALID
8236181	https://phishtank.org/phish_detail.php?phish_id=8236180... added on Jul 24th 2023 5:29 AM	by dms	INVALID
8229529	https://scontent.fmaa10-1.fna.fbcdn.net/ added on Jul 19th 2023 2:02 AM	by lhernandez	INVALID
8225101	https://synthesisgroup.es/ added on Jul 15th 2023 1:26 AM	by asd123456	INVALID
8224545	https://www.jetbrains.com/ru-ru/pycharm/download/... added on Jul 14th 2023 7:25 PM	by Felix0101	INVALID
8224082	https://thehackernews.com/2023/06/anatsa-banking-trojan-targeting-user... added on Jul 14th 2023 9:06 AM	by Felix0101	INVALID
8221373	https://202.32.243.204/ added on Jul 12th 2023 7:37 AM	by dms	INVALID
8221338	https://phishtank.org/phish_detail.php?phish_id=8221337... added on Jul 12th 2023 6:41 AM	by dms	INVALID
8220780	http://boughtbymany.us6.list-manage.com/track/click?u=5ae23627d87422b0... added on Jul 11th 2023 8:02 PM	by Josecitosgirl	INVALID
8220686	https://sis.redsys.es/sis/p2f?t=BB8442B0627CA302E7F7BF78F2B842D0E2E963... added on Jul 11th 2023 5:56 PM	by BPhy	INVALID

Figure 7. PhishTank Data Feed.

For invalid phishes or benign URLs, PhishTank does not have them in an available file. Therefore, they had to be selectively scrapped from the phishing feed that is displayed on the website. URLs that were too long were being truncated by the website and had to be scrapped from the specific sub-directories to obtain the URLs.

9.2. PhishStats

PhishStats has a public CSV feed with phishing URLs from the past 30 days that is updated every 90 min. The CSV file was downloaded, and the data were parsed in the desired format, as depicted in Figure 8.

	A	B	C	D	E
1	#####				
2	# PhishScore PhishStats		#		
3	# Score ranges: 0-2 likely 2-4 suspicious 4-6 phishing 6-10 omg phishing!		#		
4	# Ranges may be adjusted without notice. List updated every 90 minutes. Do not crawl #				
5	# too much at the risk of being blocked.		#		
6	# Many Phishing websites	so keep that #			
7	# in mind before blocking	without any warrant.	#		
8	# CSV: Date	Score	URL	IP	
9	#####				
10	5/4/2023 5:43	6.4	https://maildinshaackckjnw428.firebaseio.com/	199.36.158.100	
11	5/4/2023 5:43	5.7	https://maildinshaackckjnw427.web.app/	199.36.158.100	
12	5/4/2023 5:43	5.9	https://maildinshaackckjnw427.firebaseio.com/	2620:0:890::100	
13	5/4/2023 5:43	6.2	https://maildinshaackckjnw426.web.app/	199.36.158.100	
14	5/4/2023 5:43	6.4	https://maildinshaackckjnw426.firebaseio.com/	199.36.158.100	
15	5/4/2023 5:43	6.2	https://maildinshaackckjnw425.web.app/	199.36.158.100	
16	5/4/2023 5:43	6.4	https://maildinshaackckjnw425.firebaseio.com/	2620:0:890::100	
17	5/4/2023 5:43	5.7	https://maildinshaackckjnw424.web.app/	199.36.158.100	
18	5/4/2023 5:43	5.9	https://maildinshaackckjnw424.firebaseio.com/	2620:0:890::100	
19	5/4/2023 5:43	6.2	https://maildinshaackckjnw423.web.app/	199.36.158.100	
20	5/4/2023 5:43	6.4	https://maildinshaackckjnw423.firebaseio.com/	199.36.158.100	
21	5/4/2023 5:42	5.7	https://maildinshaackckjnw422.web.app/	199.36.158.100	
22	5/4/2023 5:42	5.9	https://maildinshaackckjnw422.firebaseio.com/	2620:0:890::100	
23	5/4/2023 5:42	5.7	https://maildinshaackckjnw421.web.app/	199.36.158.100	
24	5/4/2023 5:42	5.9	https://maildinshaackckjnw421.firebaseio.com/	2620:0:890::100	
25	5/4/2023 5:42	5.7	https://maildinshaackckjnw420.web.app/	2620:0:890::100	

Figure 8. Downloaded CSV file from PhishStats.

9.3. OpenPhish

OpenPhish publishes its limited data feed of malicious URLs to its website, which is updated every 12 h (see Figure 9). The data that are published lists the URLs in a single column, facilitating the task of extracting the data with the aid of a scraper.



Figure 9. OpenPhish Data Feed.

9.4. Cloud Server

Due to the phishing feed of PhishStats and OpenPhish being available for only a limited amount of time, a solution was required to constantly extract data from them. Therefore, a cronjob was run every 12 h via a cloud server, which ensured that URLs that were published on these platforms could be automatically gathered. Figure 10 shows this.

```
[ec2-user@ip-172-31-22-75 ~]$ crontab -l
*/12 * * * /usr/bin/python3 /home/ec2-user/openphish.py
*/12 * * * /usr/bin/python3 /home/ec2-user/phishstats.py
```

Figure 10. Crontab in Cloud Server.

An AWS Free Tier cloud server was used to complete this task as a single instance of a Linux server could be run 24/7, without incurring additional costs. PuTTY was used to connect to the Linux server remotely and WinSCP to download the datasets.

10. Data Preprocessing

Building a good deep-learning model for malicious URL recognition requires effective data preprocessing. It guarantees that the model can efficiently be trained from the data and generate accurate predictions. Four main steps were taken to preprocess the data, namely data labeling, removing duplicate rows, removing rows with certain keywords, and feature extraction.

10.1. Data Labeling

Data labeling is the basis of supervised learning, a popular technique for developing deep learning models. The model can learn to differentiate between malicious and benign URLs with the help of properly labeled data, which serves as the foundation for accurate predictions. The URLs were labeled zero if they were benign and one if they were malicious. This was performed based on the data source which the URL originates from. The data were later consolidated to form a single CSV file with the “URL” and “malicious” columns.

10.2. Removing Duplicate Rows

Duplicate rows may appear due to the data being collected from different sources and may contain the same URL. When training the deep learning model to detect malicious URLs, duplicate rows in the dataset can cause biases and inefficiencies. As the model will give the repeated occurrences excessive weight, duplicate rows have the potential to inject bias into the training data and result in an unbalanced representation of patterns. Furthermore, duplicate rows may cause the model to overfit to the point that it memorizes the duplicates rather than learning patterns that apply to other cases. Consequently, the model may function well on training data but poorly on actual data. Figure 11 shows the script used for this.

```
import pandas as pd

csv_file_path = 'datasets/input.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(csv_file_path)

# Check for duplicate rows based on the "URL" column
duplicates = df[df.duplicated(subset='URL', keep=False)]

# Remove the duplicate rows
df = df.drop_duplicates(subset='URL', keep=False)

# Save the updated DataFrame back to a CSV file
df.to_csv('datasets/output.csv', index=False)
```

Figure 11. Script to Remove Duplicate Rows.

10.3. Removing Rows with Redacted Keywords

Redacted keywords are words or phrases that have been purposefully hidden or masked, usually to safeguard private data. Examples of redacted keywords include “redacted”, “ionos” and email@example.com (see Figure 12). Most of the redacted keywords were found to be email addresses which were obfuscated to hide the original email address. Redacted keywords can occur because of data sensitivity, non-disclosure agreements, or privacy issues, which reduces the dataset’s usefulness for deep learning model training. It may be difficult for the model to efficiently find patterns if redacted keywords affect how other features in the data are interpreted. Therefore, rows containing redacted keywords were removed completely from the dataset. Figure 13 shows the script for this.

5931	http://365bet9.vip/			
5932	http://viabpc.com/			
5933	http://unescomedlab.unirc.it/mo6/index.html			
5934	https://Zooo.softart.com.br/#redacted@abuse.ionos.com			
5935	https://www.jibenmensek.top/			

Figure 12. Example of Row with Redacted Keyword.

```
import pandas as pd

# Define the input and output file paths
input_file_path = 'datasets/input.csv'
output_file_path = 'datasets/output.csv'

# Define the strings to be removed
strings_to_remove = ['redacted']

# Read all columns from the CSV file into a DataFrame
df = pd.read_csv(input_file_path)

# Convert the "URL" column to strings
df['URL'] = df['URL'].astype(str)

# Filter the DataFrame to remove rows containing specified strings in the "URL" column
filtered_df = df[~df['URL'].str.lower().str.contains('|'.join(strings_to_remove))]

# Save the entire DataFrame (including all columns) to a new CSV file
filtered_df.to_csv(output_file_path, index=False)
```

Figure 13. Script to Remove Rows with Redacted Keywords.

10.4. Feature Extraction

Only lexical features were used for the deep learning model’s training. The host-based and content-based features were purposefully left out because of the difficulties and time limits involved in extracting them. Websites are dynamic and subject to quick changes. A once benign URL might later turn malicious or stop working. The host-based and content-based features extracted from URLs would require manual monitoring and verification, and this process would take up a huge amount of time.

Although URLs are often represented as text data, deep learning models require numerical input. By transforming URLs into numerical representations that capture the crucial elements of each URL, feature extraction fills up this gap. The model can learn and recognize patterns linked to URLs with the aid of feature extraction, which improves the accuracy of detection. Table 2 shows the 32 features that were chosen to be extracted from the dataset that was collected.

Table 2. List of Feature Descriptions.

Feature	Description
URL Length	The total length of the URL string.
Hostname Length	The length of the hostname part of the URL.
Path Length	The length of the path part of the URL.
Digit Count	The count of numeric digits in the URL.
Alphabet Count	The count of alphabetic characters in the URL.
Subdomain Count	The number of subdomains in the URL.
Subdirectory Count	The number of subdirectories in the URL path.
Query Count	The count of query parameters in the URL.
Fragment Count	The count of URL fragments or anchors.
HTTP Scheme	A binary indicator of whether the URL uses 'http' or 'https'.
Is IP Address	A binary indicator of whether the URL is an IP address.
Has Port	A binary indicator of whether the URL includes a port number.
At Count	The count of the '@' symbol in the URL.
Comma Count	The count of commas ',' in the URL.
Double Slash Count	The count of double slashes '//' in the URL.
Equal Count	The count of equal signs '=' in the URL.
Hyphen Count	The count of hyphens '-' in the URL.
Percent Count	The count of percent signs '%' in the URL.
Period Count	The count of periods '.' in the URL.
Question Count	The count of question marks '?' in the URL.
Semicolon Count	The count of semicolons ';' in the URL.
Underscore Count	The count of underscores '_' in the URL.
Account Count	The count of the 'account' keyword in the URL.
Admin Count	The count of the 'admin' keyword in the URL.
Banking Count	The count of the 'banking' keyword in the URL.
Client Count	The count of the 'client' keyword in the URL.
Confirm Count	The count of the 'confirm' keyword in the URL.
Login Count	The count of the 'login' keyword in the URL.
Server Count	The count of the 'server' keyword in the URL.
Signin Count	The count of the 'signin' keyword in the URL.
Webscr Count	The count of the 'webscr' keyword in the URL.
URL Shortener	A binary indicator of whether the URL is shortened by a URL shortening service.

The extracted features were then inserted into the dataset, together with the 'URL' and 'Malicious' columns for the training of the deep learning model later. A total of 817,997 rows of URLs remained after preprocessing the data, 468,005 of which were malicious and 349,992 were benign. A sample is depicted in Table 3.

Table 3. URLs with Extracted Features.

URL	URL Length	Hostname Length	Path Length	Digit Count	Alphabet Count	Subdomain Count	Subdirectory Count	Query Count	Fragment Count	HTTP Scheme	Is IP Address	Has Port
https://www.20-minsw.com/?formCode=share-savings&productId=42&_=/application#KJWqMdlUIBn8PPpbVhTylJ/hfYJoHVq15eA/Iw== (accessed on 8 February 2024)	137	16	1	13	101	1	0	3	0	1	0	0
https://magalu.semanadasofertas.site/?category=1&refer=9169593 (accessed on 8 February 2024)	66	28	1	8	47	1	0	2	0	1	0	0
https://bufflo.csmarketplace.ink/boX9kxo1BV/9kblhk4ciw/6kuepzgmsq?q=boX9kxo1BV&s=d789f0501227e18a6936a0c7c72eb336 (accessed on 8 February 2024)	117	24	33	30	74	1	3	2	0	1	0	0
https://literate-silent-list.gltch.me/?gq=shannon.delucia@delawarepark.com (accessed on 8 February 2024)	75	30	1	0	62	1	0	1	0	1	0	0
https://info-cnft.fr/digi/ext/eml/r?par=aHR0cHM6Ly9saXRlcmF0ZS1zaWxlbmQtbGlzdC5nbGl0Y2gubWU/Z3E9c2hhbm5vbi5kZWx1Y2lhQGRIbGF3YXJlcGFyay5jb20=&emtr=11765-319138-CPujObw-2 (accessed on 8 February 2024)	173	13	15	31	123	0	4	2	0	1	0	0
https://centralqueenslandweddings.com.au/wp-content/upgrade/%D9%88%D8%AD%D8%AF%D8%A9%20%D8%A7%D9%84%D8%AA%D8%AD%D9%83%D9%85/%D9%88%D8%AD%D8%AF%D8%A9%20%D8%A7%D9%84%D8%AA%D8%AD%D9%83%D9%85/religion/%D8%B1%D8%A6%D9%8A%D8%B3%D9%8A/zimb/#.kae ss@delawarepark.com (accessed on 8 February 2024)	258	32	193	54	134	1	7	0	1	1	0	0
https://srm.dewa.gov.ae/sap/public/bc/icf/logoff?redirecturl=http://ipfs.io/ipfs/bafkreih3yloptrsyciwzo2o5sdbvf5gchtbjzi75wyj6tpdxwptmtt5ie/?af=c2FuZHJhcGVycnlAdHdpdFRlci5jb20=&p2=2019-3-1-Hyderabad-1c (accessed on 8 February 2024)	206	15	25	20	156	2	5	2	0	1	0	0
https://arweave.net/P8V058h4ADQLsx2704M6UTyyF5kY0hN_ZkFVq9AN-QE#info@cafedoranjeboom.nl (accessed on 8 February 2024)	87	11	44	12	65	0	1	0	1	1	0	0
https://s3.amazonaws.com/appforest_uf/fl680189150363x698855450976705300/in dex.html?email=accounts@centrica.com (accessed on 8 February 2024)	111	16	59	32	65	1	3	1	0	1	0	0
https://kayueglobal.com/wp-includes/certificates/.ms/index.html#nobody@mycraftmail.com&1c95e6a311558835533c61d96baa7725-787238hjsjgd893aef940f45a46ebd a60f5d152f15541-230920n8=1c95e6a311558835533c61d96baa7725-239ngKq0-a97297e 51a448db6a3955e02ebb1eb5b (accessed on 8 February 2024)	253	15	40	105	127	0	4	0	1	1	0	0
http://sainara.com.hk/img/.img/?email=nobody@mycraftmail.com&-a9d614bfb4f 9b7402a1a67002c1545c2-HDdas-cs0p271vm06y62qj14us29-h3z-3cL8kNZ9QOvbIZ4b Ad4JusrVkP8vCgm37QIM-20yvsdku-a9d614bfb4f9b7402a1a67002c1545c2 (accessed on 8 February 2024)	210	14	10	65	123	1	2	2	0	0	0	0
https://kayueglobal.com/wp-includes/IXR/index%20(11).html#nobody@mycraft mail.com&14f8024367861fe177ba566f8fcca9bc-787238hjsjgd81b82ebfc06faeb63c 499532245e28d30-230920n8=14f8024367861fe177ba566f8fcca9bc-239ngKq0-7f6412e 4051c56c4432bfc636ccdc04v (accessed on 8 February 2024)	251	15	38	106	123	0	3	0	1	1	0	0
https://www.hkplasticlymphatic.com/style/stylesheet/.css/?email=nobody@myc raftmail.com&-c02f18d8fbb045c266f2d2eac14cd5ba-HDdas-dsk8mhqay9nhkz3mka6t -h3z-0uMMUUbLcgU2NsSFy2FxAuqJ1hrldCsgzZQ4s-20yvsdku-c02f18d8fbb045c266f 2d2eac14cd5ba	234	26	23	44	167	1	3	2	0	1	0	0
https://hangfashion.com.vn/profiles/standard/translations/myid.telstra.com/mana ge/?view=logIn&appIdKey=fcd00c0656cc490&country (accessed on 8 February 2024)	134	18	56	9	105	1	5	3	0	1	0	0

Table 3. Cont.

URL	URL Length	Hostname Length	Path Length	Digit Count	Alphabet Count	Subdomain Count	Subdirectory Count	Query Count	Fragment Count	HTTP Scheme	Is IP Address	Has Port
https://kayueglobal.com/wp-includes/IXR/index%20(11).html#nobody@mycraftmail.com&ea046c47edc7aba836e0ff626708572d-787238hjsjgd8e20e664855f787d2d096095f8671f227-230920n8=ea046c47edc7aba836e0ff626708572d-239ngKq0-7236c4eb8932d7924131ec5603d1cbe9 (accessed on 8 February 2024)	251	15	38	111	118	0	3	0	1	1	0	0
http://cpmapro.ca/download/.d/?email=nobody@mycraftmail.com&-21245748eb0c5ebfd6701664de3ea53c-HDdas-bya2i7nksf8bzq389h1cm13-h3z-It7cS02gu9Kd7MBnGOOSf5SFjHLUWfUofbZX-20yvsdku-21245748eb0c5ebfd6701664de3ea53c (accessed on 8 February 2024)	210	10	13	58	131	0	2	2	0	0	0	0
https://www.ofertasamericanas.online/298589309/?console-playstation-5-ps5-+-controle-dualsense-playstation-5-+-game-uncharted-colecao-legado-dos-ladroses-ps5-em-promo%EF%BF%BD%EF%BF%BD-na-americanas&cod=255035033 (accessed on 8 February 2024)	217	28	11	22	155	1	1	2	0	1	0	0
https://doximex.vn/wp-includes/Ppou/login.php?cmd=submit_log&id=OTE1MTc0NTU3OTE1MTc0NTU3&session=OTE1MTc0NTU3OTE1MTc0NTU3 (accessed on 8 February 2024)	130	10	28	12	100	0	3	3	0	1	0	0
https://exodus-wallet.securityfixes.com/index?user=jemd@ozemail.com.au&ID=g3Wk37o5d92q2l4t407xDyJ97x1H8 (accessed on 8 February 2024)	107	31	6	15	77	1	1	2	0	1	0	0

11. URL Detection Model Architecture Specifications

The selection of the model architecture is a crucial step in the design of an accurate deep-learning system for supporting the identification of malicious URLs. A well-designed architecture can have a significant impact on the model’s ability to accurately identify threats while maintaining a low false positive rate. We developed six different models for Phishing URL detection that are based on various neural network architectures. We developed detection models based on Feedforward Neural Network (FNN), Bi-directional RNN (Bi-RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN). Performance-related details on these architectures are discussed in Section 12.

The efficiency of the developed neural network models is measured using standardized measurement matrices of confusion matrix, accuracy, precision, recall (sensitivity), and F1-score. They provide a comprehensive measure of the quality of the machine learning model.

11.1. Confusion Matrix

The confusion matrix is defined as depicted in Figure 14.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 14. Confusion Matrix.

TP (True Positive): The model correctly predicted a positive class.

FN (False Negative): The model incorrectly predicted a negative class when it should have been a positive one.

FP (False Positive): The model incorrectly predicted a positive class when it should have been a negative one.

TN (True Negative): The model correctly predicted a negative class.

11.2. Accuracy

The model’s accuracy indicates how frequently its predictions are accurate. It is determined by dividing the number of correctly predicted instances, which include true positives and true negatives by the total number of cases.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

11.3. Precision

Precision measures how many of the total number of predictions that are specified as positive are correctly assigned. The ratio of actual (true) positives to all cases that were predicted to be positive is used to compute it.

$$\text{Precision} = TP / (TP + FP)$$

11.4. Recall (Sensitivity)

Recall or Sensitivity is the total number of actual positive cases that were predicted correctly. The ratio of actual (true) positives to the overall number of positive occurrences is used to compute it.

$$\text{Recall (Sensitivity)} = \text{TP} / (\text{TP} + \text{FN})$$

11.5. F1 Score

Precision and Recall are balanced by the F1 Score, which aids in determining the right trade-off for the given situation. It offers a balance between these two measures, which is beneficial in situations when datasets are unbalanced.

$$\text{F1 Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

12. Architecture Performance Comparisons

Six different neural network architectural models were developed and compared: Feedforward Neural Network (FNN), Bi-directional RNN (Bi-RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN). These models were chosen based on previous studies, which have shown that these models have demonstrated promising performance in a variety of phishing detection-related applications [22–26].

The training and testing workflow for the models are as follows:

1. Preprocessed dataset discussed in Sections 8 and 9 is used;
2. Dataset is split into a training set and an evaluation set at an appropriate ratio;
3. Training dataset is used to develop the six deep-learning models;
4. Evaluation dataset is used to test the efficiency of the models;
5. Matrices specified in Section 11 are used for the evaluation of the models;
6. Execution time is measured for each model to evaluate the performance speed.

For instance, Figure 15 is the code representation for FNN implementation. Performance comparisons among the different deep learning architectures are illustrated in Table 4.

Table 4. Performance Comparisons of Deep Learning Architectures.

Architecture	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
Bi-RNN	0.94284	0.96712	0.93166	0.94906	[67,128 2961] [6390 87,121]	1886.289
FNN	0.925097	0.92400	0.94682	0.93527	[62,807 7282] [4972 88,539]	210.8156
GRU	0.93113	0.93312	0.94740	0.94021	[63,740 6349] [4918 88,593]	248.538
LSTM	0.93052	0.93429	0.94490	0.93956	[63,875 6214] [5152 88,359]	321.976
RNN	0.92371	0.92708	0.94049	0.93374	[63,172 917] [5564 87,947]	253.741
CNN	0.94333	0.95843	0.94169	0.94999	[66,270 3819] [5452 88,059]	629.896

12.1. Results Discussion

The CNN architecture model was selected due to its overall performance in the evaluation measured and fast training times. The initial CNN architecture model contains multiple layers, and its architecture is explained with reference to the code in Figure 16:

Input layer: The input layer defines the model's input shape. The last dimension, one, denotes that there is only one feature per time step, while the first value shows the number of features in the input data. This is typical for time series or sequence data.

```

# Get the start time
start_time = time.time()

# Read and split the data
xNumeric, x_train, x_test, y_train, y_test = read_and_split_data('datasets/MixedURLsDataset.csv')

# Define the MLP (Multi-Layer Perceptron) model architecture
input_layer = keras.Input(shape=(xNumeric.shape[1],))
dense_layer_1 = keras.layers.Dense(128, activation='relu')(input_layer)
dropout_1 = keras.layers.Dropout(0.5)(dense_layer_1)
dense_layer_2 = keras.layers.Dense(64, activation='relu')(dropout_1)
dropout_2 = keras.layers.Dropout(0.5)(dense_layer_2)
dense_layer_3 = keras.layers.Dense(32, activation='relu')(dropout_2)
output_layer = keras.layers.Dense(1, activation='sigmoid')(dense_layer_3)

model = keras.Model(inputs=input_layer, outputs=output_layer)

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train, epochs=10, batch_size=64, validation_data=(x_test, y_test))

# Evaluate the model
evaluate_model(model, x_test, y_test)

# Calculate execution time
calculate_execution_time(start_time)

# Save the trained model
model.save('models/fnn.keras')

```

Figure 15. FNN Architecture Design.

Conv1D layers: The convolutional layer utilizes the Rectified Linear Unit (ReLU) activation function and includes filters, each with a kernel size of three. To extract features from the input data, convolutional layers are used.

Max-pooling layers: This layer, which uses max-pooling with a pool size of two, is inserted after the convolutional layer. Max-pooling shrinks the feature maps' spatial size while preserving crucial data.

Flatten layer: This layer transforms the 3D tensor output of the previous layer into a 1D tensor. This step is necessary to connect the convolutional layers to the fully connected layers.

```

# Define the CNN model architecture
input_layer = keras.Input(shape=(xNumeric.shape[1], 1))
conv1 = keras.layers.Conv1D(filters=32, kernel_size=3, activation='relu')(input_layer)
maxpool1 = keras.layers.MaxPooling1D(pool_size=2)(conv1)
conv2 = keras.layers.Conv1D(filters=64, kernel_size=3, activation='relu')(maxpool1)
maxpool2 = keras.layers.MaxPooling1D(pool_size=2)(conv2)
flatten = keras.layers.Flatten()(maxpool2)
dense1 = keras.layers.Dense(128, activation='relu')(flatten)
output_layer = keras.layers.Dense(1, activation='sigmoid')(dense1)

model = keras.Model(inputs=input_layer, outputs=output_layer)

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train, epochs=10, batch_size=64, validation_data=(x_test, y_test))

```

Figure 16. Initial CNN Architecture Design.

Dense layer: This layer has 128 units, a ReLU activation function, and it is fully linked. Fully connected layers make decisions based on the features that the convolutional layers have extracted.

Output layer: One dense neuron with a sigmoid activation function makes up the output layer. It produces a probability score that ranges from zero to one, which is the usual range for binary classification issues.

Model compilation: The Adam optimizer, a popular option for optimizing neural networks, is used in the model. Binary classification jobs are well suited by the loss function, which is set to Binary Cross-Entropy. The disparity between the expected and real labels is quantified. Accuracy will be reported by the model as a measure to monitor throughout training.

The number of convolutional and max-pooling layers has been increased to further increase accuracy, as depicted in Table 5. Four convolutional and max-pooling layers have been selected as the accuracy increased minimally with an increase in execution time.

Table 5. Performance Comparisons among a range of Convolution Layers.

Architecture	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
2 Conv1D and max-pooling layers	0.94333	0.95843	0.94169	0.94999	[66,270 3819] [5452 88,059]	629.896
3 Conv1D and max-pooling layers	0.946937	0.96358	0.94279	0.95307	[66,757 3332] [5349 88,162]	795.653
5 Conv1D and max-pooling layers	0.94676	0.97074	0.93504	0.95256	[67,454 2635] [6074 87,437]	964.9478
4 Conv1D and max-pooling layers	0.94699	0.96456	0.94187	0.95308	[66,853 3236] [5435 88,076]	936.976

A dropout layer was placed in between the flatten and dense layers. Randomly changing a portion of the input units to zero during each training stage helps minimize overfitting. It serves as a type of regularization and might motivate the network to learn stronger features. The results of adding the dropout layers with different dropout rates were measured and compared to those without the dropout layer as depicted in Table 6. Not adding a dropout layer still yielded better results but took a longer time to train.

Table 6. Performance Comparisons among different Dropout Rates.

Architecture	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
With 0.2 dropout rate	0.94452	0.95610	0.946391	0.95122	[66,026 4063] [5013 88,498]	706.858
With 0.5 dropout rate	0.94671	0.96424	0.94168	0.95283	[66,824 3265] [5453 88,058]	689.515
With 0.7 dropout rate	0.94500	0.97129	0.93131	0.95088	[67,515 2574] [6423 87,088]	700.178
Without dropout layer	0.94699	0.96456	0.94187	0.95308	[66,853 3236] [5435 88,076]	936.976

12.2. Activation Function

In the context of neural networks, an activation function introduces non-linearity to the model. It aids the network in discovering intricate relationships and patterns within the data, empowering it to make precise predictions. ReLU, Softmax, Leaky ReLU, Parametric ReLU (PReLU), Exponential Linear Unit (ELU), and ThresholdedReLU have been tested in the development of the architecture as depicted in Table 7. Using the ELU activation function was found to have the highest accuracy, with a slight decrease in execution time.

Table 7. Performance Comparisons among different Activation Functions.

Activation Function	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
ReLU	0.94699	0.96456	0.94187	0.95308	[66,853 3236] [5435 88,076]	936.976
Softmax	0.92949	0.93394	0.94337	0.93863	[63,850 239] [5295 88,216]	865.938
Leaky ReLU	0.94577	0.96861	0.93543	0.95173	[67,255 2834] [6038 87,473]	740.610
PReLU	0.94328	0.96450	0.93519	0.94962	[66,871 3218] [6060 87,451]	901.384
ThresholdedReLU	0.88110	0.86350	0.94067	0.90043	[56,185 13,904] [5548 87,963]	673.648
ELU	0.94795	0.97006	0.93788	0.95370	[67,383 2706] [5808 87,703]	800.498

12.3. Optimizer

The goal of optimization, which entails reducing or maximizing an objective function, is at the heart of machine learning. The objective of supervised learning is to reduce the size of a loss function that measures the discrepancy between model predictions and actual data. This minimization procedure comprises iteratively modifying the model’s parameters to find the optimal setting that most closely fits the data.

The process of adjusting parameters is propelled by optimizers. They decide how each training iteration updates the model parameters. Optimizers are essentially in charge of exploring the parameter space to arrive at a setup that minimizes the loss function. The method by which optimizers operate is to compute the gradients of the loss function about the parameters of the model. These gradients show which way the parameters should be changed to lower the loss. The Adam, Nadam, and Stochastic Gradient Descent (SGD) optimizers were used and compared. The results can be seen in Table 8.

Table 8. Performance Comparisons among different Optimizers.

Optimizer	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
SGD	0.93629	0.94017	0.94892	0.94453	[64,443 5646] [4776 88,735]	841.052
Nadam	0.94665	0.96549	0.94027	0.95271	[66,947 3142] [5585 87,926]	821.842
Adam	0.94795	0.97006	0.93788	0.95370	[67,383 2706] [5808 87,703]	800.498

12.4. Loss Function

The model can iteratively modify its parameters during training to decrease loss with the help of loss functions, which act as a quantifiable measure of this inaccuracy. Loss functions essentially act as a guide for the learning process, measuring how well the model is doing. Various loss functions have been tested to see which has the highest accuracy and Binary Cross-Entropy has an advantage, as depicted in Table 9.

12.5. Class Imbalance

Oversampling and under-sampling are approaches used to handle class imbalance concerns in the context of machine learning and classification tasks, with the main objective of enhancing model accuracy. When working with datasets where one class considerably outnumbers the other, skewing model performance, these techniques are helpful.

Table 9. Performance Comparisons among Different Loss Functions.

Loss Function	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
Mean Squared Error	0.94589	0.97089	0.93332	0.95174	[67,473 2616] [6235 87,276]	789.436
Mean Squared Logarithmic Error	0.94559	0.96170	0.94233	0.95192	[66,580 3509] [5392 88,119]	735.121
Binary Cross-Entropy	0.94795	0.97006	0.93788	0.95370	[67,383 2706] [5808 87,703]	800.498

Class imbalance happens when there are disproportionately more instances of one class than the other. This frequently leads to the model in binary classification having a high bias towards the dominant class. As a result, the model may not perform well for the minority class, producing predictions with low accuracy, poor generalization, and bias.

By raising the proportion of members of the minority class, oversampling is a method for balancing the distribution of classes. The oversampling technique known as Synthetic Minority Over-Sampling (SMOTE) was applied. The class distribution is balanced by creating synthetic samples for the minority class. On the other hand, by choosing a subset of examples at random, under-sampling is a strategy that lowers the number of instances in the majority class. The imbalanced-learn library was used to balance the data. The imbalanced data may still have higher accuracy as compared to the oversampled and under-sampled data. This could be because noise can be introduced in the dataset by oversampling or under-sampling. In our example, under-sampling or oversampling does not have much impact on the results compared to the original dataset that was used, as illustrated in Table 10.

12.6. Batch Size

The amount of data samples that the neural network processes in a single forward and backward pass during training is referred to as the batch size. To train deep learning models, especially neural networks, on huge datasets, it is more feasible to divide the data into batches. A trade-off between computing efficiency, training stability, and convergence speed must be made while determining the optimal batch size.

Table 10. Comparison between Different Balances of Data.

Data	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
Oversampling	0.94792	0.97164	0.93621	0.95359	[67,534 2555] [5965 87,546]	832.164
Under-sampling	0.94624	0.96605	0.93893	0.95230	[67,004 3085] [5710 87,801]	649.639
Imbalanced	0.94795	0.97006	0.93788	0.95370	[67,383 2706] [5808 87,703]	800.498

It can be computationally demanding to process huge datasets all at once, and they could not fit into the training hardware’s memory. Utilizing resources effectively is made possible via batching. During training, batching introduces noise that may have a regularization impact that keeps the model from overfitting to the training set of data. Table 11 shows the performance comparisons over different batch sizes. We conclude that a batch size of 128 works best for this dataset.

Table 11. Performance Comparisons of Different Batch Sizes.

Batch Size	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
32	0.94221	0.94875	0.95021	0.94948	[65,290 4799] [4655 88,856]	1103.911
64	0.94795	0.97006	0.93788	0.95370	[67,383 2706] [5808 87,703]	800.498
256	0.94602	0.96081	0.94407	0.95237	[66,489 3600] [5230 88,281]	465.139
128	0.94857	0.96575	0.94348	0.95449	[66,961 3128] [5285 88,226]	549.7330

12.7. Epoch

During a neural network’s training phase, an epoch is a single trip of the complete training dataset. A neural network is usually trained over several epochs, during which the model iteratively adjusts its parameters in response to the training data to enhance performance. Using 12 epochs was found to be the optimal setting, which resulted in the highest accuracy while maintaining fast execution times, as depicted in Table 12.

Table 12. Performance Comparisons over Different Epochs.

Epochs	Accuracy	Precision	Recall	F1 Score	Confusion Matrix	Execution Time (s)
5	0.94506	0.96583	0.93704	0.95121	[66,989 3100] [5887 87,624]	273.181
10	0.94857	0.96575	0.94348	0.95449	[66,961 3128] [5285 88,226]	549.7330
13	0.94832	0.96463	0.94420	0.95431	[66,852 3237] [5217 88,294]	789.818
15	0.94663	0.96721	0.93844	0.95261	[67,114 2975] [5756 87,755]	761.121
20	0.94711	0.96676	0.93978	0.95308	[67,068 3021] [5631 87,880]	996.013
12	0.94908	0.96946	0.94055	0.95479	[67,319 2770] [5559 87,952]	618.987

In summary, the combination of artificial intelligence with phishing detection and training has been both a challenging and promising undertaking. The use of GPT-3, a state-of-the-art language model, for the twin purposes of phishing URL detection and convincing phishing campaign generation, was a noteworthy feature of this research. As the URL detection sub-system uses a neural network, future improvements will depend significantly on available new training datasets. This use of cutting-edge artificial intelligence highlighted how such technologies can be used in a scalable way to deal with practical problems in a specified context. It also emphasizes how the cybersecurity landscape is always changing, especially when faced with attacks of progressively increasing sophistication.

13. Conclusions

In this research, we have developed a prototype solution that integrates both phishing attack detection and end-user phishing education. The use of large language models, like GPT-3 and a Convolutional Neural Network Deep Learning Architecture, enables the prototype to detect email-based phishing attacks and allows phishing campaigns customized to specified business use cases to be generated for end-user training.

By maximizing the defense effectiveness of phishing security at a granular level, AI and generative AI technologies could significantly reduce overall cyber risk. To comprehen-

sively mitigate increasingly sophisticated email phishing attacks therefore, businesses and their employees must both understand how cybercriminals may be using the technology, as well as embrace it as part of an evolving hybrid security framework.

Author Contributions: Conceptualization, P.K.K.L. and A.Z.Y.L. with oversight from Cyber Security Agency of Singapore; methodology, A.Z.Y.L.; software, A.Z.Y.L.; validation, A.Z.Y.L., P.K.K.L. and V.B.; formal analysis, A.Z.Y.L.; investigation, A.Z.Y.L.; resources, A.Z.Y.L.; data curation, A.Z.Y.L.; writing—original draft preparation, P.K.K.L.; writing—review and editing, P.K.K.L., A.Z.Y.L. and V.B.; visualization, P.K.K.L.; supervision, P.K.K.L.; project administration, P.K.K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Training datasets for phishing detection are extracted from public domain phishing link repositories detailed in Section 9 of this paper.

Acknowledgments: We thank the Cyber Security Agency of Singapore for their guidance and advice during the development of the research prototype.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. SlashNext, The State of Phishing 2023. Available online: <https://slashnext.com/wp-content/uploads/2023/10/SlashNext-The-State-of-Phishing-Report-2023.pdf> (accessed on 26 December 2023).
2. Griffiths, C. The Latest 2023 Phishing Statistics (Updated December 2023). Available online: <https://aag-it.com/the-latest-phishing-statistics/> (accessed on 27 December 2023).
3. Basit, A.; Zafar, M.; Liu, X.; Javed, A.R.; Jalil, Z.; Kifayat, K. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommun. Syst.* **2020**, *76*, 139–154. [CrossRef] [PubMed]
4. U.S. Department of Health and Human Services, Health Sector Cybersecurity Coordination Center (HC3), AI-Augmented Phishing and the Threat to the Health Sector, *White Paper*, Report: 202310261200, 26 October 2023. Available online: <https://www.hhs.gov/sites/default/files/ai-and-phishing-as-a-threat-to-the-hph-white-paper-tpclear.pdf> (accessed on 28 December 2023).
5. Mirsky, Y.; Demontis, A.; Kotak, J.; Shankar, R.; Gelei, D.; Yang, L.; Zhang, X.; Pintor, M.; Lee, W.; Elovici, Y.; et al. The Threat of Offensive AI to Organizations. *Comput. Secur.* **2023**, *124*, 103006. [CrossRef]
6. Jackson, K.A. A Systematic Review of Machine Learning Enabled Phishing. *arXiv* **2023**, arXiv:2310.06998.
7. Lim, E.; Tan, G.; Hock, T.K.; Lee, T. *Turing in a Box: Applying Artificial Intelligence as a Service to Targeted Phishing and Defending against AI-generated Attacks*; GovTech: Singapore, 2021. Available online: <https://i.blackhat.com/USA21/Wednesday-Handouts/US-21-Lim-Turing-in-a-Box-wp.pdf> (accessed on 28 December 2023).
8. Deloitte Risk Advisory, Phishing as a Service. June 2018. Available online: <https://www2.deloitte.com/content/dam/Deloitte/in/Documents/risk/in-ra-phishing-as-a-service-noexp.pdf> (accessed on 21 December 2023).
9. Jawahar, M.G.; Abdul-Mageed, L.V.S. Lakshmanan, Automatic Detection of Machine Generated Text: A Critical Survey. *November arXiv* **2020**, arXiv:2011.01314.
10. Seymour, J.; Tully, P. Generative Models for Spear Phishing Posts on Social Media, NIPS Workshop on Machine Deception. *arXiv* **2018**, arXiv:1802.05196.
11. Begou, N.; Vinoy, J.; Duda, A.; Korczy, M. Exploring the Dark Side of AI: Advanced Phishing Attack Design and Deployment Using ChatGPT. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Orlando, FL, USA, 2–5 October 2023.
12. Falade, P.V. Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2020**, *9*, 185–198. [CrossRef]
13. Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.* **2020**, *30*, 681–694. [CrossRef]
14. Karanjai, R. Targeted Phishing Campaigns using Large Scale Language Models. *arXiv* **2022**, arXiv:2301.00665.
15. McGuffie, K.; Newhouse, A. The Radicalization Risks of GPT-3 and Advanced Neural Language Models. *arXiv* **2020**, arXiv:2009.06807.
16. Akin, F.K. The Art of CHATGPT Prompting: A Guide to Crafting Clear and Effective Prompts. Available online: <https://fka.gumroad.com/1/art-of-chatgpt-prompting> (accessed on 1 January 2024).
17. Akin, F.K. F/awesome-CHATGPT-Prompts: This Repo Includes CHATGPT Prompt Curation to Use CHATGPT Better. GitHub. Available online: <https://github.com/f/awesome-chatgpt-prompts> (accessed on 1 January 2024).
18. Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. *arXiv* **2023**, arXiv:2302.04023.
19. Joshi, P.; Santy, S.; Budhiraja, A.; Bali, K.; Choudhury, M. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. *arXiv* **2020**, arXiv:2302.04023.
20. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [CrossRef]

21. Alkaiissi, H.; McFarlane, S.I. Artificial Hallucinations in ChatGPT: Implications in Scientific Writing. *Cureus* **2023**, *15*, 2. [[CrossRef](#)] [[PubMed](#)]
22. Rakotoasimbahoaka, A.C.; Randria, L.; Razafindrakoto, N.R. Malicious URL detection Using majority vote method with machine learning and deep learning models. In Proceedings of the 2020 International Conference on Interdisciplinary Cyber Physical Systems (ICPS), Chennai, India, 28–29 December 2020; IEEE: Piscataway, NJ, USA; pp. 37–43.
23. Crişan, A.; Florea, G.; Halasz, L.; Lemnaru, C.; Oprisa, C. Detecting malicious URLs based on machine learning algorithms and word embeddings. In Proceedings of the 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 3–5 September 2020; IEEE: Piscataway, NJ, USA; pp. 187–193.
24. Mourtaji, Y.; Bouhorma, M.; Alghazzawi, D.; Aldabbagh, G.; Alghamdi, A. Hybrid rule-based solution for phishing URL detection using convolutional neural network. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–24. [[CrossRef](#)]
25. Yang, P.; Zhao, G.; Zeng, P.P. Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access* **2019**, *7*, 15196–15209. [[CrossRef](#)]
26. Wei, W.; Ke, Q.; Nowak, J.; Korytkowski, M.; Scherer, R.; Woźniak, M. Accurate and fast URL phishing detector: A convolutional neural network approach. *Comput. Netw.* **2020**, *178*, 107275. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.