applied system innovation

MDPI

Article

Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails

Molly Dewis and Thiago Viana *

Cyber and Technical Computing, University of Gloucestershire, Cheltenham GL50 2RH, UK; mollydewis@glos.ac.uk

* Correspondence: tviana1@glos.ac.uk

Abstract: Using technology to prevent cyber-attacks has allowed organisations to somewhat automate cyber security. Despite solutions to aid organisations, many are susceptible to phishing and spam emails which can make an unwanted impact if not mitigated. Traits that make organisations susceptible to phishing and spam emails include a lack of awareness around the identification of malicious emails, explicit trust, and the lack of basic security controls. For any organisation, phishing and spam emails can be received and the consequences of an attack could result in disruption. This research investigated the threat of phishing and spam and developed a detection solution to address this challenge. Deep learning and natural language processing are two techniques that have been employed in related research, which has illustrated improvements in the detection of phishing. Therefore, this research contributes by developing Phish Responder, a solution that uses a hybrid machine learning approach combining natural language processing to detect phishing and spam emails. To ensure its efficiency, Phish Responder was subjected to an experiment in which it has achieved an average accuracy of 99% with the LSTM model for text-based datasets. Furthermore, Phish Responder has presented an average accuracy of 94% with the MLP model for numerical-based datasets. Phish Responder was evaluated by comparing it with other solutions and through an independent *t*-test which demonstrated that the numerical-based technique is statistically significantly better than existing approaches.

Keywords: phishing; spam; deep learning; machine learning; natural language processing

Citation: Dewis, M.; Viana, T. Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails. *Appl. Syst. Innov.* **2022**, *5*, 73. https://doi.org/10.3390/asi5040073

Academic Editor: Dorota S. Temple

Received: 13 June 2022
Accepted: 26 July 2022
Published: 28 July 2022

1. Introduction

Since the beginning of the COVID-19 pandemic, remote working has become more popular, and email continues to be used as one of the main forms of communication. Due to this, phishing and spam emails are still prevalent. Phishing is a technique used to deceive users into providing their personal information; spam is where uninvited emails are sent to users either with the intention of promoting products and services or, likewise to phishing, gain information from the user; benign emails are non-malicious emails received on a daily basis. There are certain traits that individuals and organisations can have that expose them to phishing and spam emails including, but not limited to, a lack of awareness around the identification of malicious emails, explicit trust, and the lack of basic security controls. The dangers and problems that phishing and spam emails can present is that credentials and personal or financial data can be stolen and unauthorised access to a network can be gained. Although there is a human element to phishing and spam detection, this research focuses on a technological approach to detecting phishing and spam emails. There are various existing solutions that use natural language processing (NLP) or machine learning (ML) to mitigate the threat of phishing and spam. For example, Ding et al. [1] have proposed a spear phishing email detection solution that uses ML algorithms such as Random Forest and Decision Tree. Additionally, Banu et al. [2] use both NLP and ML to deduce whether an email is phishing. However, this research proposes a detection solution that uses deep learning and natural language processing to accurately detect phishing and spam emails.

2. Literature Review

This section concerns literature related to the problem addressed in Section 1. It is divided into Section 2.1 concerning the definitions of phishing and spam, Section 2.2 discussing the elements a detection solution should integrate, and Section 2.3 highlighting the research gap.

2.1. Definitions of Phishing and Spam

As previously mentioned, phishing intends to deceive users and can be used as a technique to steal information or gain unauthorised access to a network. On the other hand, spam is typically nuisance emails that in many cases will be sent immediately to the junk or spam folder. Marková et al. [3] categorise both spam and phishing as types of malicious emails which can be detected using machine learning.

Current research demonstrates that there are many ways of delivering phishing and spam, but via email is the most common. This is discussed by Priestman et al. [4] who state that email is a targeted form of communication which corresponds with the threat of phishing where individuals or organisations are targeted by threat actors, hoping to inherit personal information from their victims. Even though Junnarkar et al. [5] mainly focus on spam emails, they highlight that the rate of exchanging information via emails is significantly increasing. Bountakas, Koutroumpouchos and Xenakis [6] illustrate that this could be a result of the COVID-19 pandemic in recent years, necessitating organisations to work remotely and threat actors proceeding to use phishing and spam emails. Therefore, this research aims to explore phishing and spam.

The make-up of a phishing or spam email depends on the sender. Banu et al. [2] outline examples of phishing emails such as fake invoices and purchase orders or security alerts and account suspension. Many threat groups distributing phishing emails will employ their own techniques to hook individuals, whereas spam emails tend to be generalised and less personal. Phishing emails will typically contain URLs or attachments, as well as employing certain words encouraging the user to believe what is written.

Regarding the consequences of phishing emails, they can act as a gateway to additional threats which Egozi and Verma [7] have illustrated, as they state phishing emails have been used to deploy file-encrypting ransomware such as CryptoLocker. Furthermore, Walkowski [8] outlines that the MITRE framework, which highlights threat actor behaviour, recognises that threat actors tend to use a spearphishing link or an attachment to gain initial access into an organisation's digital estate. The consequences of spam emails can slightly differ to phishing emails in that companies may be fined if spam emails are sent to customers. In the case of American Express, they were fined for sending 4 million nuisance emails to customers who had opted out [9], thus outlining that spam emails can be sent to users for promotion purposes and not necessarily to gain information from them.

Overall, as both phishing and spam still present themselves as a threat, this research aims to provide a detection solution for detecting both phishing and spam emails.

2.2. Existing Detection Solutions

In this section, a discussion on the key elements of a phishing detection solution is presented.

2.2.1. Research Methodology

Using machine learning for any task requires a clear research methodology to ensure the research is valid and replicable. Replicability means researchers can reproduce and modify current solutions when tackling any problem. This is addressed by both AbdulNabi and Yaseen [10] and Junnarkar et al. [5] who provide flowcharts of their methodologies, outlining the steps that should be conducted when developing a detection solution. Phishing and spam research should have a clear methodology to ensure the most suitable techniques are used in an efficient manner.

Datasets are crucial for the development of phishing and spam email detection solutions. One characteristic of a dataset is its source. Addressing this, Bountakas, Koutroumpouchos and Xenakis [6] used the Enron email corpus and the Jose Nazario phishing corpus, as they both have sample sizes sufficient for training and testing detection solutions. Although both these datasets provide phishing and spam emails which can be used to train a detection solution, collating phishing emails may have to be done by a researcher as Ding et al. [1] have outlined that phishing emails can contain sensitive information. Regardless of the dataset being used to train a detection solution, Nass, Levit and Gostin [11] have illustrated that incomplete datasets will impact research and so datasets need to be handled carefully.

Another characteristic is whether the dataset is balanced or unbalanced. Using an unbalanced dataset can heavily impact the performance of a classifier as seen in research by Ding et al. [1] who had to apply the KM-SMOTE algorithm to lessen the impact of their unbalanced dataset. As the whole purpose of developing a phishing detection solution is to reduce human involvement, it seems logical to ensure all datasets are balanced so no results are inaccurate. Therefore, the weighting of a dataset must be taken into consideration when conducting research because an inaccurate detection solution as a result of the size of a dataset could lead to phishing emails successfully disrupting organisations.

Additionally, splitting the dataset into training and testing data is a key characteristic of machine learning research. Despite Alhogail and Alsabih [12] suggesting that two-thirds of a dataset should be used for training, the ideal size of a dataset has not been provided and so two-thirds of a small dataset might not be sufficient for training a detection solution. Marková et al. [3] and Ding et al. [1] outline how much of their dataset is used for training and testing, yet they are not clear about the number of benign and phishing emails in each dataset, which should be considered because if the training dataset just contained benign emails, the detection solution would not have been properly trained. Therefore, splitting a dataset into training and testing data will ensure that the detection solution can be sufficiently trained and tested to identify phishing and spam emails.

2.2.2. Feature Extraction

One method of identifying phishing or spam emails is by extracting features. Most research seems to understand the implications of clicking on a URL or opening an attachment—two features synonymous with phishing and spam emails. Alhogail and Alsabih [12] state that the body of an email is the key to identifying a malicious email, as it can contain useful features. One feature that Salloum et al. [13] believe should be extracted is URLs and this would be sensible as Ding et al. [1] found that 90% of attackers used URLs in their research. It seems illogical that Aggarwal, Kumar and Sudarsan [14] ignored emails containing links in their research, especially when Toulas [15] illustrates that a malicious link was used to trick users in a recent RuneScape-themed phishing campaign. Likewise, Ding et al. [1] extracted the number of attachments and their file types from spear phishing emails, thus illustrating that information about attachments should be extracted. Montalbano [16] further outlines that a recent malicious email campaign prompts users to download a PDF which will result in the propagation of the Snake Keylogger malware. Therefore, detection solutions should extract certain features from emails, alongside implementing NLP techniques.

2.2.3. Natural Language Processing

NLP is about the processing of language and one popular technique amongst related literature is TF-IDF. Despite Bountakas, Koutroumpouchos and Xenakis [6] achieving more than 90% accuracy for all ML algorithms when used alongside TF-IDF to understand word frequency and the importance of a word, they did not consider email addresses and domains, which are useful characteristics. An important factor for addressing the phishing problem was considered by Stojnic, Vatsalan and Arachchilage [17] who used TF-IDF to observe the techniques used by attackers. Using TF-IDF to understand the types

of words and their importance could be a useful NLP technique to use for the identification of phishing and spam emails.

Another NLP technique often used is tokenization. Alhogail and Alsabih [12] use tokenization to separate text into individual words which is an important task because they emphasise that analysing the text features of an email is a vital research area. Despite Banu et al. [2] failing to outline how they conducted tokenization, they highlight that it is a useful method for isolating malicious keywords in an email which could have benefited Marková et al. [3] who do not appear to have used tokenization. Moreover, Stojnic, Vatsalan and Arachchilage [17] illustrate that attackers often use words that invoke urgency or reward and words that concern urgency and trust in an email's subject. Therefore, tokenization would highlight the words and phrases that typically appear in phishing and spam emails.

Older research by Aggarwal, Kumar and Sudarsan [14] and Verma, Shashidhar and Hossain [18] employed techniques such as part of speech and word stemming; however, it appears other research has opted for newer techniques such as BERT. AbdulNabi and Yaseen [10] used a BERT-based model to consider the context of words in spam emails, achieving an accuracy of 98.67% and although spam and phishing are different, it seems understanding the context of words is a useful approach. NLP techniques that produce inaccurate results could result in spam or phishing emails bypassing detection solutions as highlighted by Bountakas, Koutroumpouchos and Xenakis [6] who used BERT with Naive Bayes which only achieved 66.54% accuracy. This emphasises that using newer NLP techniques that produce accurate results should be a priority for any detection solution, especially since phishing and spam emails, if left undetected, can lead to organisations being left vulnerable.

2.2.4. Deep Learning

Deep learning is one type of machine learning that can be used to detect phishing and spam emails. As outlined by Sathya, Premalatha and Suwathika [19], deep learning is where the machine learns from unstructured data such as emails without supervision. Many researchers have moved onto deep learning algorithms for detecting phishing and spam emails, as opposed to machine learning algorithms which have been extensively discussed in previous research. AbdulNabi and Yaseen [10] used convolutional neural network (CNN) for sentence classification, emphasising that it can be useful in spam and phishing detection. Consequently, Salloum et al. [13] specify that future phishing detection research should focus on deep learning techniques such as recurrent neural networks (RNN) and CNNs as fewer detection solutions use these techniques. However, Yang et al. [20] illustrate that deep learning-based detection can include CNN, RNN, RCNN and DNN models.

The usefulness of deep learning is addressed by Lavanya and Sasikala [21] who illustrate that deep learning techniques reveal the masked patterns and find meaningful information. Additionally, Yang et al. [20] outline that deep learning can effectively overcome the need for manual feature extraction. Therefore, deep learning should be used in the development of detection solutions to ensure meaningful information can be gathered from phishing and spam emails.

2.3. Research Gap

Phishing and spam detection is not a new research area; however, detection solutions developed to tackle these types of emails have extensively focused on machine learning approaches. Additionally, existing research has either focused on solely phishing or spam. Therefore, there are gaps in the existing literature, and it is indicated that hybrid approaches encompassing deep learning and NLP, as well as catering for text-based and numerical-based datasets, can become the way forward.

3. Materials and Methods

This section concerns the materials and methods used within this research. It is divided into Section 3.1 concerning the datasets used in this research, Section 3.2 regarding the features extracted and Section 3.3 illustrating the experiments conducted.

3.1. Datasets

Table 1 illustrates the datasets used within this research to ensure the models worked and could effectively achieve a high accuracy and precision, suitable for phishing and spam detection, regardless of the dataset being used.

Table 1. Datasets.

Dataset	Source	Text or Numerical
Spambase [22]	UCI Machine Learning Repository [23]	Numerical
Phishing Email Collection [24]	Kaggle	Numerical
Email Spam Dataset—Spam Assassin [25]	Kaggle	Text
Spam Email [26]	Kaggle	Text
Spam Classification for Basic NLP [27]	Kaggle	Text
Email Spam Classification Dataset CSV [28]	Kaggle	Numerical

Each dataset was split into 70% for training (and validation) and 30% for testing. The UCI Spambase [22], which contains a collection of spam emails, was used as there can be similarities between spam and phishing emails. Verma and Gautam [29] used the UCI Spambase dataset, achieving 98.413% in phase 1 of their research and further improving their results when applying various classification algorithms. Although the Kaggle datasets did not appear to be well-known datasets, they were open-source datasets that contained features found in both phishing and spam emails. Additionally, the datasets in Table 1 were chosen because they each included benign and spam or phishing data.

3.2. Feature Extraction

Deep learning has been used to complete both feature extraction and classification. The numerical-based datasets were reliant on the author for the attributes that featured in the dataset, whereas the text-based datasets consisted of mainly email bodies which were cleaned and transformed using NLP techniques. Table 2 outlines the features/attributes within each dataset.

Table 2. Feature Extraction.

Dataset	Features
Spambase [22]	Word frequency, character frequency, total number of capital letters, uninterrupted sequences of capital letters and the label.
Phishing Email Collection [24]	Total number of characters, vocabulary richness, account, access, bank, credit, click, identity, inconvenience, information, limited, minutes, password, recently, risk, social, security, service, suspended, total number of function words, unique word, and phishing status.
Email Spam Dataset—Spam Assassin [25]	Email body which includes URLs and label.
Spam Email [26]	Category and message.
Spam Classification for Basic NLP [27]	Category and message—raw text messages including URLs—plain messages with headers and HTML tags.
Email Spam Classification Dataset CSV [28]	3000 most common words and the label.

3.3. Architectural Experimentation

This section illustrates the two experiments that were conducted for determining how the phishing detection solution was to be developed using Python. Python was the language chosen because it had been used previously by the authors and there are a variety of libraries that can be implemented. The development of the phishing detection solution was used on the same computer with the specifications outlined in Table 3.

Table 3. Computer specifications.

Component	Specification
Processor	11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80 GHz 2.80 GHz, 4 cores
RAM	16.0 GB
Operating System	Windows 10 Home Version 21H2
Networking	Intel(R) Wireless-AC 9461
GPU	Intel(R) Iris(R) Xe Graphics
Python	Python 3.9 (64-bit)
PyCharm	PyCharm Community Edition 2021.2.1

Figure 1 illustrates the experiments that were conducted, as well as the performance metrics that were analysed for each experiment.

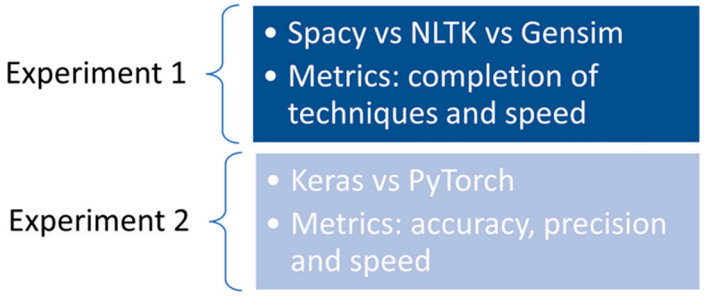

Figure 1. Experiments.

The experiments allowed for deep learning and NLP-based Python libraries to be compared, determining which were the most suitable for the research problem. NLP was used to ensure that the dataset was suitably prepared for training and deep learning gave the detection solution the opportunity to accurately classify emails as either benign or phishing/spam. Xiao et al. [30] use deep learning for detecting phishing websites as it generally produces better classification results. Furthermore, Lauriola, Lavelli and Aiolli [31] have stated that deep learning boosts the performance of NLP applications.

3.3.1. Natural Language Processing

The first experiment concerned natural language processing (NLP) as NLP techniques were useful for the data pre-processing phase. The Spam Classification for Basic NLP dataset from Kaggle [27] was used for the NLP experiment. This dataset contained spam messages which included useful features such as IP addresses and URLs, and it was a text-based dataset so various NLP techniques could be experimented with.

There were three Python libraries considered for ascertaining the most suitable library for this research problem: NLTK, Spacy and Gensim. As accuracy and precision could not be used as performance metrics for the NLP experiment, two factors were considered when electing the most suitable Python library for NLP-based tasks. The first factor was how many of the chosen NLP techniques each Python library could complete, and the second factor was how quickly the techniques could be applied to the dataset.

In a similar approach to Banu et al. [2] and Alhogail and Alsabih [12], this research used tokenization as its main NLP technique. Tokenization was used to split the dataset

into tokens, which made it easier to observe whether words synonymous with phishing and spam emails were present in the dataset. Bountakas, Koutroumpouchos and Xenakis [6] identified POS tagging as a task that facilitated lemmatization, which was considered the core of the pre-processing task whereas Banu et al. [2] and Verma Shashidhar and Hossain [18] used stemming to process text in emails. Removing stopwords has been widely used as a technique in phishing and spam detection research but Egozi and Verma [7] highlight that they keep stopwords in their research as they consider them a fundamental part of an email. Therefore, stopwords and uppercase text have been kept in this research as they provide context in emails and Bagui et al. [32] highlight that context can be essential for phishing detection. Tokenization, stemming, lemmatization and POS tagging were the techniques used to determine the most suitable Python library.

As NLTK was able to quickly conduct all the NLP techniques used during the experiment, NLTK was to be used for the NLP-related tasks in this research.

3.3.2. Deep Learning

The aim of the second experiment was to establish whether Keras or PyTorch was the most suitable Python library for deep learning. The UCI Spambase dataset [22] was used as it provided a dataset ideal for binary classification. Figure 2 outlines a basic design of the deep learning model that was used for the deep learning experiment.

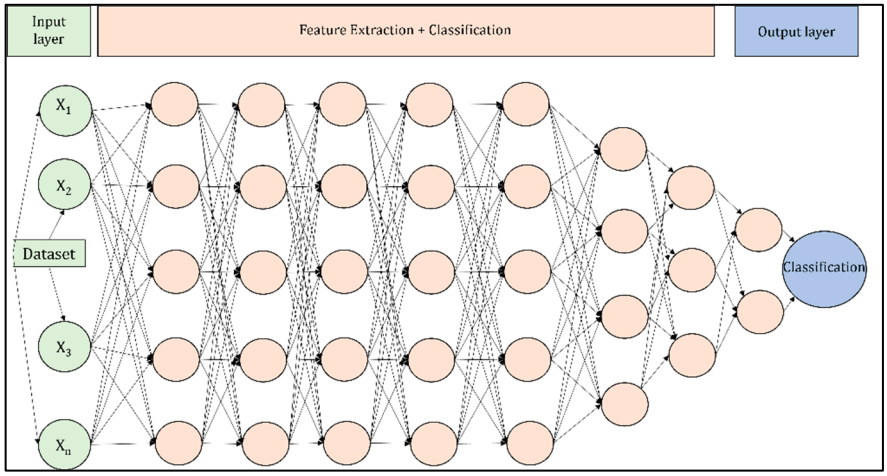

Figure 2. Deep learning model design.

To establish whether Keras or PyTorch was the more suitable deep learning library, accuracy and precision were the performance metrics considered (see Table 4).

Table 4. Performance metrics.

Metric	How Was It Measured?
Accuracy	How accurately the detection solution detects phishing/spam emails.
Speed	Added the start and end time to the detection solution to understand the time it takes to detect phishing/spam emails.
Precision	Determine the proportion of emails that are phishing/spam to those that were detected as phishing/spam [6].

The opportunity to identify the optimal parameters for this research arose when experimenting with Keras (see Figure 3).

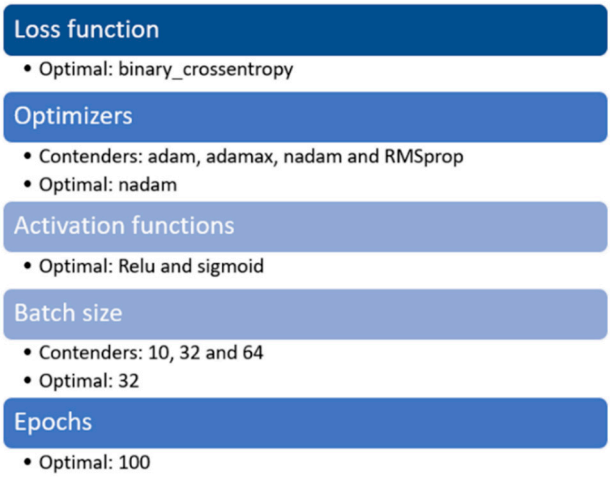

Figure 3. Optimal parameters.

Binary_crossentropy was the loss function used because Teja, Sasank and Reddy [33] state that it works best for binary classification problems such as phishing detection. Figure 4 outlines various optimizers which produced accuracies higher than 90%. It was expected that the *nadam* optimizer would be used as Pavan Kumar, Jaya and Rajendran [34] achieve a high accuracy with this optimizer; however, during the Keras experiment, the nadam optimizer achieved the highest accuracy. Kewei et al. [35] in their research into fraud detection (a binary classification problem) used the *nadam* optimizer.

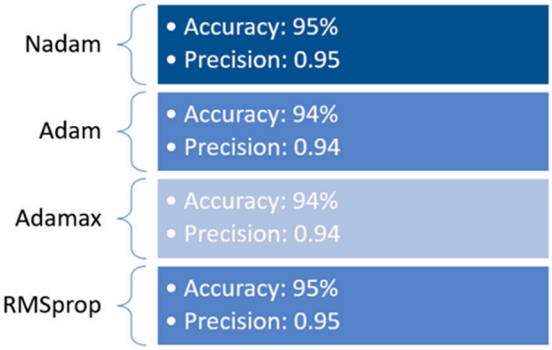

Figure 4. Keras optimizers.

Relu and sigmoid were the activation functions considered, as when combined with the optimizers illustrated in Figure 3, high accuracies on average were achieved (see Table 5 below).

Table 5. Optimizers and activation functions.

Optimizers	Input	Hidden Layers (×7)	Output (×2)	Average Accuracy (from 5 Runs)
Nadam	Relu	Relu	Sigmoid	86%
Adam	Relu	Relu	Sigmoid	81%
adam	Sigmoid	Sigmoid	Sigmoid	94%
Adamax	Sigmoid	Sigmoid	Sigmoid	94%
Nadam	Sigmoid	Sigmoid	Sigmoid	94%
RMSprop	Sigmoid	Sigmoid	Sigmoid	94%
Nadam	Relu	Sigmoid	Sigmoid	95%
RMSprop	Relu	Sigmoid	Sigmoid	94%
adam	Relu	Sigmoid	Sigmoid	94%
adamax	Relu	Sigmoid	Sigmoid	93%

Bagui et al. [32] found that relu was the most effective activation function but they used sigmoid for the end Dense layer as it performs well for binary classification. Although Butt et al. [36] achieved 92% for classifying phishing URLs when using relu for the input and hidden layers and sigmoid for the output layer, these results were not replicated in this research. Using the relu activation function for the input and hidden layers produced a 60% accuracy which is not suitable for phishing detection. Therefore, the green row in Table 3 illustrates the optimal parameters for the model used during the experimentation phase of this research.

After choosing the activation functions, the batch size was tested. Do et al. [37] illustrate that a batch size of 32 is optimal for all deep learning algorithms, hence a batch size of 32 was used. However, to clarify that this batch size was optimal, a batch size of 10 and 64 were also tested as Shabudin et al. [38] and Xiao et al. [30] both used a batch size of 10 for classifying phishing websites. Over five runs, a batch size of 32 produced an average accuracy of 93.78%, whereas a batch size of 64, although faster, achieved an average accuracy of 92.93%. Therefore, a batch size of 32 alongside 100 epochs was used for the rest of the deep learning experiment.

Once the optimal parameters in Figure 5 had been chosen, these parameters were tested on another dataset.

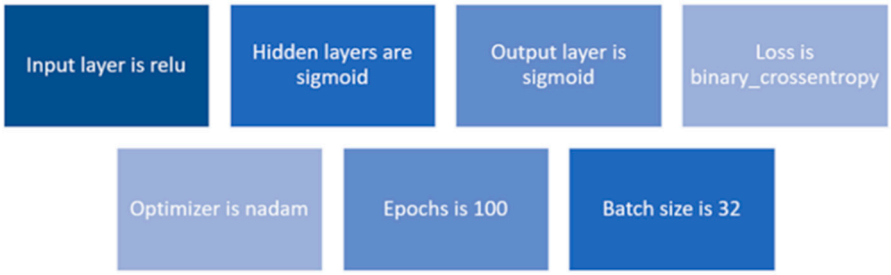

Figure 5. Optimal parameters.

Despite the Kaggle dataset by Akashsurya156 and Kul [24] taking longer to classify due to its size, an accuracy of 98% and a precision of 0.98 was achieved, which suggested that the speed of the detection solution depended on the dataset being used.

The optimal model parameters were used for the PyTorch experiment, along with the model in Figure 2.

Despite both Keras and PyTorch being able to produce high accuracies and precisions, PyTorch produced lower accuracies than Keras (see Figure 6); therefore, it was determined that Keras would be used in the overall detection solution.

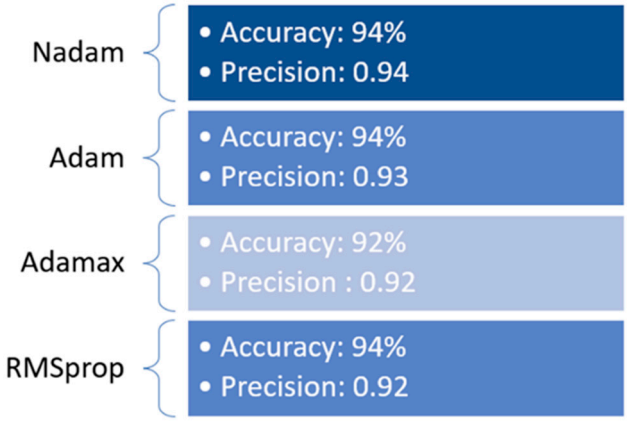

Figure 6. PyTorch optimizers: accuracy and precision.

For determining the most suitable deep learning algorithms, each dataset was split into a training and testing dataset. Using a similar approach as Bagui et al. [32], 70% of each

dataset collated was used for training (and validation) and 30% for testing; however, each dataset was kept separate so they could be treated as variables for the detection solution. Table 6 highlights that the quantity of benign and phishing/spam values was balanced within the training and testing datasets as Bountakas, Koutroumpouchos and Xenakis [6] draw attention to using balanced datasets.

Table 6. Training and testing datasets.

Phishing Email Collection Dataset [24]	UCI Spambase Dataset [22]
Training: 5845 phishing and 5845 benign Testing: 1268 spam and 1268 benign	Training: 2505 phishing and 2505 benign Testing: 544 spam and 544 benign

Overall, the training datasets were used for determining the most suitable deep learning algorithms.

A comparison of algorithms to determine the most suitable algorithm for this research problem was conducted (see Table 7).

Table 7. Algorithms.

Numerical-Based Dataset	Text-Based Dataset
MLP Simple RNN LSTM	Simple RNN LSTM

Existing research tends to opt for a combination of algorithms for spam and phishing detection. Ghourabi, Mahmood and Alzubi [39] used both CNN and LSTM, achieving 98% accuracy, whereas Sriram et al. [40] demonstrate that CNN, when used for image classification, can achieve up to 99% accuracy. Although CNN is known for image classification, McGinley and Monroy [41] demonstrated that their CNN model was able to classify real-world phishing emails, achieving an accuracy of 98%. However, CNN was disregarded as an algorithm that could be used to classify numerical-based datasets as it did not work well with the datasets used.

MLP was one of the algorithms tested during the experimentation phase. Figure 7 displays all the instances used to test the suitability of MLP for this research problem and the green rows outline the highest accuracies achieved.

Run 1	Run 2	Run 3	Run 4	Run 5	Average accuracy	Model specification
89.99	95.23	94.17	94.88	88.64	92.584	Input and 5 of the hidden layers have 5 nodes and then number of nodes decrement by 1 until the output layer.
93.5	94.6	95.11	84.79	92.2	92.04	Start at 10 nodes in a layer and decrement by 1 until the output layer.
88.1	49.98	89.79	85.46	50.02	72.67	Each layer decrements by 1, before each dense layer there is a dropout and flatten layer but not on the input layer.
91.09	89/67	93.97	94.72	94.76	92.842	Decrementing by 1 for each layer, as well as having a dropout and flatten just before the output layer.

Figure 7. MLP experiment results highlighting the best average accuracies.

It was found that with 50 epochs and using all the optimal parameters in Figure 7, MLP on average achieved a relatively high accuracy, although not when using the dataset from Kaggle by Akashsurya156 and Kul [24] which only achieved a 50% accuracy.

As Lee et al. [42] claim that RNN along with LSTM are some of the most widely used deep-learning techniques in recent classification studies, both Simple RNN and LSTM were tested. Accuracy using Simple RNN was seen to be temperamental using the UCI Spambase dataset by Hopkins et al. [22] as an accuracy of 75% and 10% was achieved across the five runs. When using the Phishing Email Collection dataset from Kaggle [24], the accuracy also differed. For phishing and spam detection, these accuracies are not good enough and as a result, this algorithm was disregarded.

The same varied accuracies were also produced when using LSTM. The implications of achieving varied accuracies in phishing detection are that an undetected phishing or spam email could severely impact organisations. Combining LSTM and Simple RNN did not see high accuracies either; however, accuracies above 70% were achieved more consistently.

Overall, MLP was found to be the most suitable algorithm for numerical-based datasets as accuracies of above 90% were achieved.

As Das et al. [43] consider Simple RNN and LSTM in their effectiveness for classifying URLs as malicious or benign, both algorithms were also tested for classifying text-based datasets. The Spam Classification dataset from Kaggle [27] was used for this part of the experimentation phase because NLP techniques and deep learning algorithms could be applied.

Using a similar approach to Gualberto et al. [44] and Bountakas, Koutroumpouchos and Xenakis [6], TF-IDF was used because it is a popular technique for classifying phishing emails. In the case of Bountakas, Koutroumpouchos and Xenakis [6], a high accuracy was achieved when using TF-IDF alongside machine learning algorithms; therefore text-based datasets were converted into numerical form using TF-IDF prior to using the deep learning model.

As previously highlighted, RNN algorithms have solely been tested for classifying the text-based datasets as Vinayakumar et al. [45] state that RNN has obtained good performance in artificial intelligence tasks including natural language processing. LSTM was the first RNN-based algorithm that was tested, producing an accuracy of 98% and precision of 0.98 consistently throughout the five runs. Even when the model was changed to resemble the number of layers defined in Figure 2, the accuracy and precision remained the same. Likewise, Simple RNN achieved an accuracy of 98% and a precision of 0.98. As both algorithms produced high accuracies and precisions, another text-based dataset was used to determine the reliability of the results. The Spam Email dataset from Kaggle by Qureshi [26] was split into a training and testing dataset and then the training dataset was used. As this second dataset was larger, only 5 epochs were used but both Simple RNN and LSTM still achieved an accuracy of 99.89% and a precision of 0.998883. Both algorithms were combined to understand their value together, but high accuracies and precisions were still achieved. As LSTM has appeared more in similar research, this algorithm was chosen for classifying text-based datasets.

It was observed from the experiments that NLTK and Keras were the most suitable Python libraries for this phishing and spam detection solution. MLP and LSTM were chosen to classify numerical-based datasets and text-based datasets, respectively. All these elements have been combined in the development of a phishing and spam detection solution.

4. Phish Responder

This section outlines the finalised elements of the Phish Responder detection solution.

4.1. Overall Structure

As Barik et al. [46] have indicated that security experts prefer command line tools for their familiarity and power and Almeida et al. [47] outline that speed is an advantage of command line tools, Phish Responder has been developed for the command line. This

allows users to choose which type of dataset they want to classify quickly and accurately (see Figure 8).

```
Detection solution start time Fri May 13 09:42:55 2022
 ---------- Phish Responder ---------------
Choose the type of dataset you want to classify:
t = Text-based Dataset
n = Numerical-based Dataset
i = Individual email
Enter selection: t
You have chosen to upload a Text-based Dataset
Input the file path for your chosen dataset when asked
Enter file path: SpamAssassin.csv
File name: SpamAssassin.csv
PhishResponder.py:36: DtypeWarning: Columns (9,10,11,12,13,14,15,16,17,18,19,20
,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44) have
mixed types. Specify dtype option on import or set low_memory=False.
  text_df = pd.read_csv(text_dataset)
Stage 1: Data Preprocessing
Beginning data preprocessing: Fri May 13 09:43:08 2022
Tokenization complete - Stemming complete
Lemmatization complete
POS Tagging complete
Data Preprocessing finished: Fri May 13 09:43:08 2022


Stages 2 and 3: Feature Extraction and Classification
shape is (99, 37)
shape is (99, 37)
Note: Output number = number of attributes in dataset
What is the output?37
2022-05-13 09:43:28.808139: W tensorflow/stream_executor/platform/default/dso_l
```

Figure 8. Command line.

4.2. Data Pre-Processing

For Phish Responder, it was found that the best approach was to only use tokenization and TF-IDF in regard to NLP techniques. Recent research such as AbdulNabi and Yaseen [10] and Alhogail and Alsabih [12] has not opted for techniques such as stemming, lemmatization, and POS tagging. These three NLP techniques were initially implemented but did not improve the accuracy of the LSTM model for text-based datasets. Therefore, these techniques were removed from the model. However, as it is previously mentioned that context is beneficial in phishing detection, these techniques were still implemented but the output was written to an external file instead in case further context was required, especially as Mishra, Shaikh and Sanyal [48] recognise that POS tagging can be used to understand the context of any phrase. Overall, these were the steps taken to achieve data pre-processing.

4.3. Feature Extraction and Classification

As previously discussed, each of the datasets used was split into training (and validation) and testing data.

Based on the results, the deep learning model used in the experimentation phase was modified and LSTM was the algorithm used for text-based datasets (see Figure 9 for the model architecture).

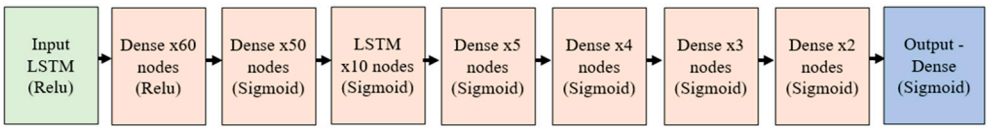

Figure 9. LSTM model for text-based datasets.

As previously mentioned, due to the impact on accuracy, the locations of the activation functions were modified and relu was used for the input layer and the first hidden layer, and sigmoid was used for the rest of the layers in the model. Overall, this is how the model for text-based datasets was developed.

Phish Responder allows the user to process numerical datasets using an MLP model (see Figure 10) that closely resembles the model used in the experiments. As the deep

learning experiment implied that achieving a good accuracy when dealing with a large dataset was more probable when adding a few more Dense layers, further Dense layers were added in the final MLP model to reduce the impact of the sudden drop in parameters between the input and hidden layers.

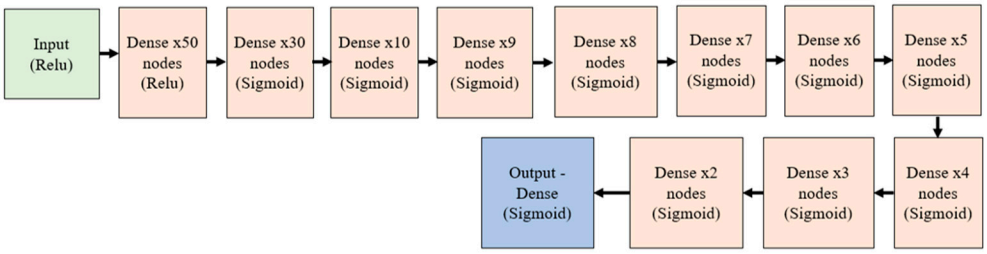

Figure 10. MLP model for numerical-based datasets.

When integrating the MLP model into the overall detection solution, the accuracy dramatically decreased when using the numerical-based datasets previously used. It was found that the datasets being used were too small; due to that, the accuracy increased significantly when using a larger numerical-based dataset. Thus, this implies that a large dataset is required for achieving a suitable accuracy in phishing and spam detection.

4.4. Individual Emails

In order to make Phish Responder more of a unique detection solution, the third option allows users to quickly extract features such as URLs, email addresses, and IP addresses from an individual email, as well as words synonymous with phishing and spam emails. This is important particularly as Ding et al. [1] used VirusTotal in their research to analyse URLs, IPs and domains found in emails. Figure 11 illustrates the extraction of features from an individual spam email from the Spam Classification for Basic NLP dataset by Naidu [27].

Figure 11. Individual email option.

This option in Phish Responder allows for further analysis of the extracted features to take place, thus potentially aiding the incident response process.

5. Evaluation and Discussion

This section evaluates Phish Responder and is divided into Section 5.1 which provides a comparison of Phish Responder with other solutions, Sections 5.2 and 5.3 which outline the strengths and limitations of Phish Responder and Section 5.4 which discusses the *t*-test.

5.1. Comparison

Phish Responder has been compared with two solutions (see Table 8).

Table 8. Research comparison.

Bountakas, Koutroumpouchos and Xenakis [6]	Junnarkar et al. [5]	Phish Responder
NLP and ML for phishing email detection. Uses TF-IDF with Logistic Regression, Decision Tree, Random Forest, GBT and Naïve Bayes. Uses the Enron email corpus and Jose Nazario's phishing corpus.	Classify spam emails using NLP techniques and ML algorithms such as Naïve Bayes and SVM. Uses the SMS Spam Collection Dataset and the Enron Spam dataset.	NLP and deep learning for phishing and spam email detection. Uses LSTM for text based and MLP for numerical based datasets. Provides an individual email element to extract interesting features. Uses the Spam Assassin from the Email Spam Dataset from Kaggle [25] and the Email Spam Classification Dataset from Kaggle [28].
Accuracies for balanced datasets using TF-IDF: LR—92.48%, DT—90.17%, RF—93.41%, GBT—91.47%, NB—92.77%. Accuracies for imbalanced datasets using TF-IDF: LR—98.41%, DT—95.92%, RF—94.43%, GBT—97.28%, NB—92.05%.	Accuracies: Naïve Bayes—95.48%, SVM—97.83%.	Accuracies: LSTM—99%, MLP—94%.
Precisions for balanced datasets using TF-IDF: LR—0.9110, DT—0.9063, RF—0.9743, GBT—0.9301, NB—0.9652. Precisions for imbalanced datasets using TF-IDF: LR— 0.9031, DT—0.8655, RF—1, GBT—0.9032, NB—0.	Precisions: Naïve Bayes—95% for 0 and 97% for 1, SVM—98% for 0 and 97% for 1.	Precisions: LSTM—0.99, MLP—0.94.

The accuracy and precision have been measured in this research so the effectiveness of Phish Responder can be evaluated and compared with the solutions in Table 8. As seen in Table 8, Phish Responder can achieve high accuracies and precisions, thus illustrating its efficiency in identifying phishing and spam emails. Over five runs, the LSTM model achieved an average accuracy of 99% and the MLP model achieved an average accuracy of 94%. Phish Responder's LSTM model with the Spam Assassin dataset can achieve a higher accuracy than the techniques in Table 8. Additionally, the MLP model produces a similar accuracy to the research outlined in Table 8. Therefore, when compared with existing research, it is confirmed that Phish Responder produces suitable accuracies and precisions for phishing and spam email detection.

Alongside accuracy and precision, Phish Responder can identify phishing and spam emails in a timely manner as it can be used via the command prompt. This provides users with a quick identification of phishing and spam emails within a dataset and a quick extraction of features from an individual email. It is not known whether the existing research, presented in Table 8, has developed its techniques for the command prompt; however, Junnarkar et al. [5] provide a distinguishable feature which is real-time classification of emails—a feature neither included in Phish Responder nor Bountakas, Koutroumpouchos and Xenakis [6].

Bountakas, Koutroumpouchos and Xenakis [6] provided a comparison of NLP and ML algorithms to determine the best combination for the detection of phishing and spam emails; they achieved at least 90% accuracy when using TF-IDF. TF-IDF is a technique that Phish Responder uses, but deep learning is used, as opposed to machine learning. Both Bountakas, Koutroumpouchos and Xenakis [6] and Junnarkar et al. [5] state that deep learning would be considered as future work in their research. Junnarkar et al. [5] also used a combination of NLP and ML algorithms, achieving respectable accuracies and precisions, but their focus was on spam emails. Bountakas, Koutroumpouchos and Xenakis [6] concentrated on the detection of phishing emails. Phish Responder differs as it has been developed to identify phishing and spam emails and both spam and phishing datasets were used to train the LSTM and MLP models. Phish Responder was trained using open-source datasets, mainly from Kaggle. Junnarkar et al. [5] also used a dataset from

Kaggle, as well as the Enron dataset which Bountakas, Koutroumpouchos and Xenakis [6] used, as it is a well-known dataset for spam and phishing detection.

The research in Table 8 used more NLP techniques in their models, placing more of an emphasis on the most accurate combination of NLP techniques and machine learning algorithms. In the case of Bountakas, Koutroumpouchos and Xenakis [6], the best combination for a balanced dataset was Word2vec and Random Forest, and for an imbalanced dataset Word2vec with Logistic Regression was the better combination. Junnarkar et al. [5] found that Naïve Bayes and SVM achieved high accuracies, alongside NLP techniques including but not limited to removing HTML tags, removing special characters, removing stopwords, stemming and lemmatization, whereas Phish Responder only implemented tokenization and TF-IDF in the text-based technique, along with writing the output of stemming, lemmatization and POS tagging to an external file. It seems that the other research in Table 8 focused on datasets where NLP techniques had to be applied, whereas Phish Responder catered for numerical-based datasets which did not need to be cleaned and transformed using NLP techniques.

Overall, Phish Responder shares similarities and differences with the research outlined in Table 8.

5.2. Strengths

Regarding the strengths of Phish Responder, the LSTM and MLP models were successfully trained using a variety of datasets containing phishing and spam emails. Phish Responder can extract interesting features from individual emails. Ding et al. [1] looked at spear phishing emails and provide a solution that extracts interesting features from an email such as URLs and IP addresses. Phish Responder aimed to replicate this feature by providing an option for users to extract URLs, email addresses, IP addresses and relevant, unique words from an email. Phish Responder was also developed to cater for text-based and numerical-based datasets, as at the data collection stage, some of the datasets were either text or numerical-based.

Solutions are useful when the key findings are presented. Therefore, when using Phish Responder, the output of the solution is written to an external *.txt* file. This allows for the key findings to be accessible in case further analysis of an email is required; this may happen when responding to or investigating a cyber incident. Furthermore, Phish Responder can be used via the command prompt when the necessary Python libraries are installed (see Figure 12).

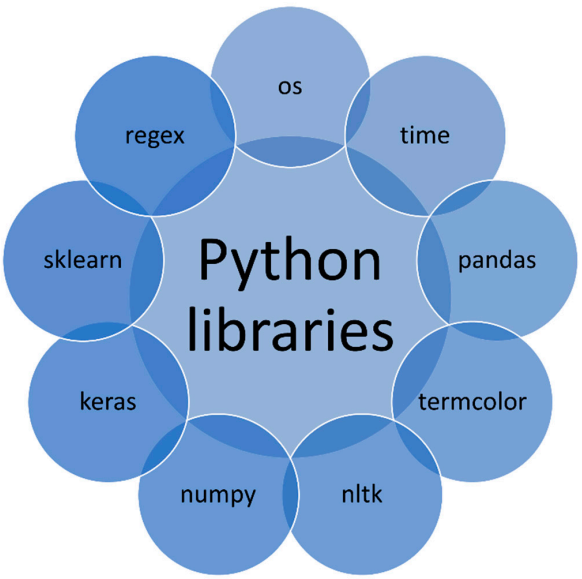

Figure 12. Python libraries.

However, Phish Responder provides accessibility, efficiency, and speed which are vital characteristics for any phishing and spam email detection solution. The quick and accurate identification of phishing and spam emails is particularly important for organisations where such malicious emails can result in business disruption if left undetected.

In summary, Phish Responder provides a quick solution to detect phishing emails, caters for various datasets and extracts interesting features from an email.

5.3. Limitations

Phish Responder could utilise more NLP techniques, as both Bountakas, Koutroumpouchos and Xenakis [6] and Junnarkar et al. [5] use at least five NLP techniques. Despite some of the known NLP techniques potentially removing necessary context from email bodies, Phish Responder does not use a wealth of NLP techniques and some of the techniques that are used are not actually employed by the LSTM model. Furthermore, although the text-based technique works and is still suitable for phishing spam detection, the *t*-test for the text-based technique could not be completed, so it cannot be determined whether it is statistically significant.

Integrating with email clients to provide real-time detection is beneficial but unlike Junnarkar et al. [5] who were successfully able to classify emails in real time, there was not enough time to develop this feature in this research. Overall, Phish Responder has weaknesses when compared to existing research.

5.4. T-Test

A *t*-test was used to compare the accuracy of the numerical-based techniques with existing research. It was hoped that the text-based technique could also be compared with existing research. Despite the text-based technique working, the statistical significance could not be determined. Table 9 displays the test conducted and the null and alternative hypotheses.

Table 9. Null and alternative hypotheses.

Test Description	Null Hypothesis	Alternative Hypothesis
Test 1: Comparing the accuracy of the numerical-based technique with existing research.	There is no statistical difference in accuracy between the numerical-based technique in this research and existing research.	There is a statistical difference in accuracy between the numerical-based technique in this research and existing research.

Using the same computer specifications as seen in Table 3, the accuracy was calculated from five runs (using fifty epochs); accuracy was the performance metric used because both Bountakas, Koutroumpouchos and Xenakis [6] and Junnarkar et al. [5] use it in their research to determine the reliability of their techniques.

As the Email Spam Classification dataset from Kaggle by Biswas [28] had been split into training and testing prior to the evaluation stage, the testing dataset was used for the *t*-test. To conduct the *t*-test, the techniques used by the other researchers were replicated, as seen in Table 10.

Table 10. Approaches for each test.

Research	Approach for Numerical-Based Datasets
Bountakas, Koutroumpouchos and Xenakis [6]	Feature selection: Chi square Classification: Logistic Regression (using an imbalanced dataset)
Junnarkar et al. [5]	SVM
Phish Responder	MLP model

Junnarkar et al. [5] achieved a higher accuracy when using SVM, as opposed to Naïve Bayes and Bountakas, Koutroumpouchos and Xenakis [6] when using an imbalanced dataset achieved the highest accuracy with Logistic Regression. Therefore, SVM and Logistic Regression were each compared with Phish Responder's MLP model.

Table 11 illustrates that Phish Responder's numerical-based technique is statistically significant and thus the null hypothesis can be rejected.

Table 11. *T*-test results.

Tests	T-Value	*p*-Value	Significant
1—Numerical-based technique vs. Junnarkar et al. [5]	10,036.88386	<0.00001	Yes (at $p < 0.05$) Yes (at $p < 0.01$)
2—Numerical-based technique vs. Bountakas, Koutroumpouchos and Xenakis [6]	10,035.26405	<0.00001	Yes (at $p < 0.05$) Yes (at $p < 0.01$)

6. Conclusions

Phish Responder is a Python-based command line solution that uses deep learning and NLP to detect phishing and spam emails. It caters for text-based and numerical-based datasets as well as individual emails. For the LSTM model for text-based datasets, an accuracy of 99% was achieved and for the MLP model for numerical-based datasets, an accuracy of 94% was achieved; both are suitable for phishing and spam detection. Interesting features such as URLs and email addresses can be extracted using the individual element of Phish Responder; such features would aid analysis in the incident response process.

Phish Responder was evaluated by conducting a *t*-test and it was determined that the numerical-based technique is statistically significant in comparison with research. Although the *t*-test could not be proven as statistically significant, it is still functional and applicable for this research problem. Additionally, Phish Responder has been compared to existing research such as Bountakas, Koutroumpouchos and Xenakis [6] and Junnarkar et al. [5] who take different approaches to Phish Responder.

Future Work

In terms of the future direction of addressing this research problem, using a combination of deep learning algorithms for phishing and spam detection is likely. Improvements on the text-based technique within Phish Responder would be made to ensure that this technique is statistically significant in comparison with similar research. Additionally, integrating Phish Responder into the incident response process would be ideal. This could be achieved by providing real-time detection of phishing and spam emails which would prevent the occurrence of further attacks.

Author Contributions: Conceptualization, M.D. and T.V.; methodology, M.D. and T.V.; software, M.D.; validation, M.D. and T.V.; writing—original draft preparation, M.D.; writing—review and editing, T.V.; supervision, T.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Ding, X.; Liu, B.; Jiang, Z.; Wang, Q.; Xin, L. Spear Phishing Emails Detection Based on Machine Learning. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 354–359. [CrossRef]
2. Banu, R.; Anand, M.; Kamath, A.; Ashika, S.; Ujwala, H.S.; Harshitha, S.N. Detecting Phishing Attacks Using Natural Language Processing and Machine Learning. In Proceedings of the 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 15–17 May 2019; pp. 1210–1214. [CrossRef]
3. Marková, E.; Bajtoš, T.; Sokol, P.; Mézešová, T. Classification of malicious emails. In Proceedings of the 2019 IEEE 15th International Scientific Conference on Informatics, Poprad, Slovakia, 20–22 November 2019; pp. 000279–000284. [CrossRef]

4. Priestman, W.; Anstis, T.; Sebire, I.G.; Sridharan, S.; Sebire, N.J. Phishing in healthcare organisations: Threats, mitigation and approaches. *BMJ Health Care Inform.* **2019**, *26*, e100031. [CrossRef] [PubMed]
5. Junnarkar, A.; Adhikari, S.; Fagania, J.; Chimurkar, P.; Karia, D. E-Mail Spam Classification via Machine Learning and Natural Language Processing. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 693–699. [CrossRef]
6. Bountakas, P.; Koutroumpouchos, K.; Xenakis, C. A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection. In Proceedings of the ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; pp. 1–12. [CrossRef]
7. Egozi, G.; Verma, R. Phishing Email Detection Using Robust NLP Techniques. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 7–12. [CrossRef]
8. Walkowski, D. MITRE ATT&CK: What It Is, How it Works, Who Uses It and Why, F5 Labs. 2021. Available online: https://www.f5.com/labs/articles/education/mitre-attack-what-it-is-how-it-works-who-uses-it-and-why (accessed on 2 November 2021).
9. Bracken, B. American Express Fined for Sending Millions of Spam Messages. 2021. Available online: https://threatpost.com/american-express-fined-spam/166412/ (accessed on 9 June 2022).
10. AbdulNabi, I.; Yaseen, Q. Spam Email Detection Using Deep Learning Techniques. *Procedia Comput. Sci.* **2021**, *184*, 853–858. [CrossRef]
11. Nass, S.J.; Levit, L.A.; Gostin, L.O. The Value, Importance, and Oversight of Health Research, Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research. 2009. Available online: https://www.ncbi.nlm.nih.gov/books/NBK9571/ (accessed on 11 November 2021).
12. Alhogail, A.; Alsabih, A. Applying machine learning and natural language processing to detect phishing email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
13. Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey. *Procedia Comput. Sci.* **2021**, *189*, 19–28. [CrossRef]
14. Aggarwal, S.; Kumar, V.; Sudarsan, S.D. Identification and Detection of Phishing Emails Using Natural Language Processing Techniques. In Proceedings of the 7th International Conference on Security of Information and Networks—SIN '14, Glasgow, UK, 9 September 2014; ACM Press: New York, NY, USA, 2014; pp. 217–222. [CrossRef]
15. Toulas, B. RuneScape Phishing Steals Accounts and in-Game Item Bank PINs, BleepingComputer. 2022. Available online: https://www.bleepingcomputer.com/news/security/runescape-phishing-steals-accounts-and-in-game-item-bank-pins/ (accessed on 9 June 2022).
16. Montalbano, E. Snake Keylogger Spreads through Malicious PDFs. 2022. Available online: https://threatpost.com/snake-keylogger-pdfs/179703/ (accessed on 10 June 2022).
17. Stojnic, T.; Vatsalan, D.; Arachchilage, N.A.G. Phishing email strategies: Understanding cybercriminals' strategies of crafting phishing emails. *Secur. Priv.* **2021**, *4*, e165. [CrossRef]
18. Verma, R.; Shashidhar, N.; Hossain, N. Detecting Phishing Emails the Natural Language Way. In *Computer Security—ESORICS 2012. European Symposium on Research in Computer Security*; Foresti, S., Yung, M., Martinelli, F., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; pp. 824–841. [CrossRef]
19. Sathya, K.; Premalatha, J.; Suwathika, S. Reinforcing Cyber World Security with Deep Learning Approaches. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; p. 766. [CrossRef]
20. Yang, R.; Zheng, K.; Wu, B.; Wu, C.; Wang, X. Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning. *Sensors* **2021**, *21*, 8281. [CrossRef]
21. Lavanya, P.M.; Sasikala, E. Deep Learning Techniques on Text Classification Using Natural Language Processing (NLP) in Social Healthcare Network: A Comprehensive Survey. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 13–14 May 2021; p. 603. [CrossRef]
22. Hopkins, M.; Reeber, E.; Forman, G.; Suermondt, J. UCI Machine Learning Repository: Spambase Data Set, UCI Machine Learning Repository. 1999. Available online: https://archive.ics.uci.edu/ml/datasets/spambase (accessed on 13 May 2022).
23. Dua, D.; Graff, C. UCI Machine Learning Repository: Citation Policy, UCI Machine Learning Repository. 2019. Available online: https://archive.ics.uci.edu/ml/citation_policy.html (accessed on 13 May 2022).
24. Akashsurya156; Kul, G. Phishing Email Collection. 2020. Available online: https://kaggle.com/akashsurya156/phishing-paper1 (accessed on 3 December 2021).
25. Nitisha. Email Spam Dataset. 2020. Available online: https://www.kaggle.com/nitishabharathi/email-spam-dataset (accessed on 1 May 2022).
26. Qureshi, F. Spam Email. 2021. Available online: https://kaggle.com/mfaisalqureshi/spam-email (accessed on 5 March 2022).
27. Naidu, C. Spam Classification for Basic NLP. 2021. Available online: https://kaggle.com/chandramoulinaidu/spam-classification-for-basic-nlp (accessed on 15 January 2022).
28. Biswas, B. Email Spam Classification Dataset CSV. 2020. Available online: https://www.kaggle.com/balaka18/email-spam-classification-dataset-csv (accessed on 5 May 2022).

29. Verma, S.; Gautam, A.K. Machine Learning Techniques for Classification of Spambase Dataset: A Hybrid Approach. In Proceedings of the ISCSIC 2019: 2019 3rd International Symposium on Computer Science and Intelligent Control, Amsterdam, The Netherlands, 25–27 September 2019; ACM: New York, NY, USA, 2019. [CrossRef]
30. Xiao, X.; Zhang, D.; Hu, G.; Jiang, Y.; Xia, S. CNN–MHSA: A Convolutional Neural Network and multi-head self-attention combined approach for detecting phishing websites. *Neural Netw.* **2020**, *125*, 303–312. [CrossRef]
31. Lauriola, I.; Lavelli, A.; Aiolli, F. An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. *Neurocomputing* **2021**, *470*, 443–456. [CrossRef]
32. Bagui, S.; Nandi, D.; Bagui, S.; White, R.J. Classifying Phishing Email Using Machine Learning and Deep Learning. In Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 3–4 June 2019; pp. 1–2. [CrossRef]
33. Teja, C.S.B.; Sasank, T.; Reddy, Y. Phishing website detection using different machine learning techniques. *Int. Res. J. Eng. Technol. (IRJET)* **2020**, *7*, 610. [CrossRef]
34. Pavan Kumar, P.; Jaya, T.; Rajendran, V. SI-BBA—A novel phishing website detection based on Swarm intelligence with deep learning. *Mater. Today Proc.* **2021**, *in press*. [CrossRef]
35. Kewei, X.; Peng, B.; Jiang, Y.; Lu, T. A Hybrid Deep Learning Model For Online Fraud Detection. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 15–17 January 2021; p. 433. [CrossRef]
36. Butt, M.H.F.; Li, J.P.; Saboor, T.; Arslan, M.; Butt, M.A.F. Intelligent Phishing Url Detection: A Solution Based On Deep Learning Framework. In Proceedings of the 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 17–19 December 2021; p. 438. [CrossRef]
37. Do, N.Q.; Selamat, A.; Krejcar, O.; Yokoi, T.; Fujita, H. Phishing Webpage Classification via Deep Learning-Based Algorithms: An Empirical Study. *Appl. Sci.* **2021**, *11*, 9210. [CrossRef]
38. Shabudin, S.; Sani, N.S.; Ariffin, K.A.Z.; Aliff, M. Feature Selection for Phishing Website Classification. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 593. [CrossRef]
39. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet* **2020**, *12*, 156. [CrossRef]
40. Sriram, S.; Sani, N.S.; Ariffin, K.A.Z.; Aliff, M. Deep Convolutional Neural Network Based Image Spam Classification. In Proceedings of the 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 4–5 March 2020; p. 115. [CrossRef]
41. McGinley, C.; Monroy, S.A.S. Convolutional Neural Network Optimization for Phishing Email Classification. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; p. 5609. [CrossRef]
42. Lee, J.; Tang, F.; Ye, P.; Abbasi, F.; Hay, P.; Divakaran, D.M. D-Fence: A Flexible, Efficient, and Comprehensive Phishing Email Detection System. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS P), Vienna, Austria, 6–10 September 2021; p. 581. [CrossRef]
43. Das, A.; Das, A.; Datta, A.; Si, S.; Barman, S. Deep Approaches on Malicious URL Classification. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–6. [CrossRef]
44. Gualberto, E.S.; De Sousa, R.T.; Vieira, T.P.D.B.; Da Costa, J.P.C.L.; Duque, C.G. The Answer is in the Text: Multi-Stage Methods for Phishing Detection Based on Feature Engineering. *IEEE Access* **2020**, *8*, 223539. [CrossRef]
45. Vinayakumar, R.; HBa, B.G.; Ma, A.K.; KPa, S. DeepAnti-PhishNet: Applying Deep Neural Networks for Phishing Email Detection CEN-AISecurity@IWSPA-2018. In Proceedings of the 1st Anti-Phishing Shared Task Pilot at 4th ACM IWSPA Co-Located with 8th ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 21 March 2018; Available online: https://www.researchgate.net/profile/M-Kumar-2/publication/326211143_DeepAnti-PhishNet_Applying_Deep_Neural_Networks_for_Phishing_Email_Detection_CEN-AISecurityIWSPA-2018/links/5d2317d5458515c11c1c15d9/DeepAnti-PhishNet-Applying-Deep-Neural-Networks-for-Phishing-Email-Detection-CEN-AISecurityIWSPA-2018.pdf (accessed on 16 March 2022).
46. Barik, K.; Das, S.; Konar, K.; Banik, B.C.; Banerjee, A. Exploring user requirements of network forensic tools. *Glob. Transit. Proc.* **2021**, *2*, 351. [CrossRef]
47. Almeida, R.; Pacheco, V.; Antunes, M.; Frazão, L. An easy-to-use tool to inject DoS and spoofing networking attacks. In Proceedings of the 2021 16th Iberian Conference on Information Systems and Technologies (CISTI), Chaves, Portugal, 23–26 June 2021; p. 2. [CrossRef]
48. Mishra, A.; Shaikh, S.H.; Sanyal, R. Context based NLP framework of textual tagging for low resource language. *Multimed. Tools Appl.* **2020**, *in press*. [CrossRef]