Replacing human input in spam email detection using deep learning.

NICHO, M., MAJDANI, F. and MCDERMOTT, C.D.

2022

Replacing Human input in Spam Email Detection using Deep Learning

Mathew Nicho[1], Farzan Majdani[2], and Christopher D. McDermott[2]

[1] College of Technology Innovation, Zayed University, Dubai, United Arab Emirates
mathew.nicho@zu.ac.ae
[2] School of Computing, Robert Gordon University, Aberdeen, United Kingdom
{c.d.mcdermott, farzan.majdani}@rgu.ac.uk

Abstract. The Covid-19 pandemic has been a driving force for a substantial increase in online activity and transactions across the globe. As a consequence, cyber-attacks, particularly those leveraging email as the preferred attack vector, have also increased exponentially since Q1 2020. Despite this, email remains a popular communication tool. Previously, in an effort to reduce the amount of spam entering a users inbox, many email providers started to incorporate spam filters into their products. However, many commercial spam filters rely on a human to train the filter, leaving a margin of risk if sufficient training has not occurred. In addition, knowing this, hackers employ more targeted and nuanced obfuscation methods to bypass in-built spam filters. In response to this continued problem, there is a growing body of research on the use of machine learning techniques for spam filtering. In many cases, detection results have shown great promise, but often still rely on human input to classify training datasets. In this study, we explore specifically the use of deep learning as a method of reducing human input required for spam detection. First, we evaluate the efficacy of popular spam detection methods/tools/techniques (freeware). Next, we narrow down machine learning techniques to select the appropriate method for our dataset. This was then compared with the accuracy of freeware spam detection tools to present our results. Our results showed that our deep learning model, based on simple word embedding and global max pooling (SWEM-max) had higher accuracy (98.41%) than both Thunderbird (95%) and Mailwasher (92%) which are based on Bayesian spam filtering. Finally, we postulate whether this improvement is enough to accept the removal of human input in spam email detection.

Keywords: Spam Detection · Phishing Emails · Simple Word Embedding · Global Max Pooling · Deep Learning

1 Introduction

Email has become a de facto standard of communication across the globe. The number of global email users, which was 3.8 billion in 2018, is set to grow to

4.48 billion by the year 2024 [67]. As the popularity of the internet continued to grow, email followed suit, resulting in users getting substantially more unsolicited emails, some of which has malicious intent and carries payloads in 'genuine' looking attachments. In this respect, spam has produced considerable economic damage [5] and is still a preferred attack vector for attackers.

While attacks vectors take multiple methods, one of the most common ways of data breach is via spam or phishing emails. Research undertaken by Verizon found that almost a quarter (22%) of data breaches were caused by impersonation, where an attacker acted as though they belonged to the company [36]. Here, the attacker leveraged email to gain the trust of a user and gather information, specially financial information. Once successful, the information was either used to commit a fraud, request further information through exploitation, or it was sold onto a third party.

By definition, spam email, also known as junk email, is any kind of unwanted, unsolicited, digital communication. Often, the email is sent out en-mass, resulting in a in reduction of Internet quality of service, and incurring considerable direct and indirect costs associated with the management of such spam [33]. Alternatively, phishing is an advanced type of spam email where the attacker spoofs genuine email and creates fraudulent websites to steal sensitive data such usernames, passwords, credit cards, and bank account details [62]. In these type of attacks, the email identity and header information are not normally verified or authenticated, such that it purports to originate from a legitimate company or bank.

Given the considerable rise in email use it is now estimated that an average person spends around 28% of a regular workweek interacting with emails. However, of the emails received, only 38% are considered relevant and important, with the rest categorised as spam [27]. In an attempt to reduce the amount of time spent on unnecessary emails many users have adopted software-based spam filters such as (*Mailwasher*[1] and *Thunderbird*[2]) which are based on bayesian statisical analysis and rely heavily upon human interaction to train the spam filter. While these bayesian-based classifiers return good accuracy, that can be further improved as more messages are classified [48], they are wholly dependent on a human completing the training task on a regular basis to remain resistant to new forms of spam.

Since the start of the Covid-19 pandemic, the rate of spam has increased, with 96% of phishing attacks now arriving by email, and a further 3% carried out through malicious websites or telephone communications (1%) [60]. The pandemic has been the driving force for a substantial increase in online meetings, activities and transactions across the globe. Armed with this knowledge, attackers have sought to explore this event circulating messages relating to Covid-19, capitalising on fear and uncertainty as the world reacted to the virus's initial outbreak progression [35]. In this respect, Covid-19-themed attacks exploded in

[1] https://www.firetrust.com/products/mailwasher-pro
[2] https://support.mozilla.org/en-US/kb/thunderbird-and-junk-spam-messages/

mid to late March 2020, linked to the Covid-19 news cycle, utilising multiple attack vectors and techniques [35].

The aim of this research is to evaluate the current state of spam detection, explore human input in the process, and evaluate the use of deep learning in this context. In doing, we seek to answer the research question: *Does the use of deep learning remove the need for human input in spam email detection?*. In this respect, first, the research evaluates popular spam and phishing detection applications (freeware software) available to the research, and used by the wider community. Second, we explore and analyse current machine learning techniques proposed for spam detection, leading to the selection of an appropriate machine learning technique for use in our study. Here, the selected machine learning technique is compared to the freeware spam filters and evaluated for accuracy and loss. Finally, we explore the use of human input during the detection process to determine whether this can be replaced by deep learning.

2 Email as an Attack Vector

Since threats to email security can come from multiple sources, it is essential to establish a comprehensive threat model based on the risk posed to a company. For example, attackers could use Traffic Distribution Systems (TDS) to effectively serve up different types of spam, and even malware, to a varying range of machines in different locations [7]. A number of protocols are used for the delivery of email, each with its own associated advantages and disadvantages. SMTP servers alone often struggle to distinguish between genuine (*ham*) and unsolicited (*spam*) email. In addition, the main drawback of sending through an SMTP server is the anonymity of a sender's identity [22]. Alternatively, IMAP can be difficult to maintain, leading to the less support and use of the protocol. Copies of messages are also stored in the server space, requiring larger amounts of mailbox space resulting in increased costs [47]. POP3 is another popular email protocol, but can consume considerable resources of a system because since messages are downloaded and saved on the local device. This introduces additional risks since if the device crashes or is stolen, data could be lost. As such, legacy email protocols like SMTP, POP3 and IMAP, are often targeted by hackers and spammers [59].

Covid-19 Related Attack Vector: As internet-worked users become more dependent on online services, they also become vulnerable to online fraud. As discussed, these threats have been accelerated since the start of the Covid-19 pandemic [1]. In this respect, the top 10 cyber security threats amid the Covid-19 pandemic were found to be DDoS attack, malicious domains, malicious websites, malware, ransomware, spam emails, malicious social media messaging, business email compromise, mobile apps, and browsing apps [39]. Of these, spam email served as a direct attack vector for malware, ransomware, business email compromise, and a supporting vector for malicious domains, and malicious websites. As such, spam was considered one of the most potent threats in the ream of online

communication. Consequently, spam emails exploiting the Covid-19 pandemic have become rampant where the most common technique deployed by spammers was *snowshoe*. This is where an attacker uses multiple IPs and domains for spam campaigns in an effort to avoid detection, where as much as 85% of emails sent were considered spam [17]. Using automated tools, the cost to reward ratio is very low, with countless emails flooding the web at negligible cost. In this respect, the FBI recently indicated that phishing campaigns had become the most common type of cybercrime in 2020, where phishing incidents nearly doubled in frequency, from 114,702 incidents in 2019, to 241,324 incidents in 2020 [68]. Close inspection showed that many were related to Covid, where the phishing attack used titles, messages contents and attachments targeted for the pandemic. These included zoom meeting requests (spoofed hyperlink), leverage of heuristic and cognitive biases such as fear to trick a user into downloaded malware embedded in remote working tools. [68]. As such the attack vector moved beyond simple spam to very specific and targeted phishing attempts [1]. Here, phishing can be considered an advanced type of spam email where phishers use spoofed emails and fraudulent websites to steal sensitive data like usernames, passwords, credit cards, and bank account details [62]. Ultimately, in this kind of targeted attack the end user has become an integral part of the overall vector leading to many researchers labelling the end user as the 'weakest link' in the security chain [28, 52]. Furthermore, beyond spam or phishing attacks, end users remain one of the most persistent vulnerabilities in many computer systems [70]. For these, and many more reasons, hackers continue to exploit human vulnerabilities rather than breaking into systems directly, ensuring spam and phishing attacks remain a real threat [29].

2.1 Spam Detection Approaches

While spam detection cannot protect a user completely, there are techniques developed by researchers and practitioners to enhance the spam detection rate. Here, we categorise the detection methods into two approaches namely: *machine learning* and *non-machine learning* approaches, with further subcategories.

Non-machine Learning Approaches: Non-machine learning methods have historically been used in spam email detection. Often, they simply include a list of email addresses or words on which the filter determines whether the given mail is spam or not [44]. Various methods have been used successfully including content based, heuristic, signature based, challenge/response and DNS blacklist. For example, content-based filtering involves automatic filtering rules to classify emails. The occurrence and distribution of words and phrases in an email are evaluated and matched against predefined rules to filter the incoming spam emails [19]. Heuristic or rule based spam filtering technique have prior rules or heuristics to assess enormous patterns which are usually regular expressions against a chosen message. The score of the message increases with similar patterns and it deducts from the score if the patterns didn't match. When the message's score outpace

a specific threshold,it is spam; else it is counted as valid [16]. Usually spammers send a replica of their spam message to all the possible email accounts they can find. When the site receives a message, it generates a signature for it and stores it in the database. To determine if the received message is spam, the anti-spam software simply checks to see if the signature for the incoming message matches with any of the signatures in the spam signature database. If it does, then the message is considered as spam [41]. Alternatively, DNS blacklisting filtering technique uses a centralised database to block all email from a specific host attempting to send the spam messages. The blacklists are static lists and require to be be maintained manually by adding entries in the database for new hosts that are considered to be spamming. The blacklist is stored and served in conjunction with the DNS system serving queries [56]. Challenge/response filtering systems send an automated reply to an email, enquiring the authenticity of the original sender in a reply, prior to delivery of the original email. The basic idea is not to block unsolicited bulk email (UBE), but rather to allow emails from humans only, who can assert that a response to challenge is needed [6].

Every spam filtering method has weaknesses that spammers can exploit and launch attacks. The limitations of main spam detection methods are summarised in Table 1.

Machine Learning Approaches: Machine learning approaches have grown in popularity and include algorithms such as support vector machine (SVM), naive bayes, k nearest neighbour and artificial neural networks. In many cases of supervised learning a set of emails which are pre-classsified (by a human) and used to train the associated ML model. This approach is more efficient to detect and tackle spams because of the machine learning system's ability to evolve itself over the time reducing the concept drift [46]. The most commonly used machine learning spam filtering is bayesian filter. Bayesian filter learns the difference between ham (non spam) and spam by looking at two categories of email message. One category is comprised of spam messages received by a site, and the other one contains ham messages received by the same site. A comparison is undertaken about how frequently a given word appears in both ham and spam messages, after which the filter determines the probability that a message containing the given word is spam [34]. Similarly, artificial neural networks perform spam filtering by either computing the rate of occurrence of keywords or patterns in the email messages. Neural network algorithms for email filtering usually attain moderate classification performance [30]. SVM filters are supervised learning models that are very potent for the identification of spam patterns and classifying them into a specific class or group. SVM is a good classifier due to its sparse data format, satisfactory recall, precision value and high classification accuracy [54].

This section demonstrates a comparison of selected Machine Learning Techniques (MLT) for identifying spam and malicious phishing emails. Studies have compared the predictive accuracy of several machine learning methods namely Logistic Regression (LR), Classification and Regression Trees (CART), Bayesian

Table 1: Spam detection methods and weaknesses

Method	Refences	Critique
Signature matching	A.Kolcz et al [40]	The spam catching rate is low. The spammer can easily avoid it. Requires frequent access to anti-spam vendor systems. Methods react to the spammer, instead of proactively rejecting spam messages.
Heuristic	Dudley et al [23]	The system has a high false-positive rate if the rules are poorly written.
Bayesian	Heron et al [34]	The system needs extremely high resource requirements. The method takes training to learn the difference between spam and ham messages.
DNS Blacklisting	Hao et al [31]	The system has a comparatively low spam catch rate.
Challenge/ Response	Alkahtani et al [6]	It has a weakness in the authentication. Wireless network security issues. If both sending and receiving mail servers implement them, dead lock will result as both servers will wait for the other to respond to their challenges.
Rule-Based System	Najadat et al [49]	The detection speed is extremely limited.
Statistical Content Filter	M.T Banday [10]	The detection speed is extremely limited

Additive Regression Trees (BART), Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNet) for predicting phishing emails [2]. S.Baadel et al considered phishing as a classification problem and outlined some of the recent intelligent machine learning techniques (associative classifications, dynamic self-structuring neural network, dynamic rule-induction, etc.) in the literature that is used as anti-phishing models [9]. Meanwhile, G.H.Lokesh et al designed a phishing classification system with the comparative study of classical machine learning techniques such as Random Forest, K nearest neighbours, Decision Tree, Linear SVC classifier, One class SVM classifier and wrapper-based features selection, which contains the metadata of URLs and use the information to determine if a website is legitimate or not [32]. Table 2 illustrates the anti-phishing tools based on machine learning algorithms.

Table 2: Anti-phishing tools based on machine learning algorithms

Tool	**Machine Learning models**	**Authors**
PHISH-SAFE	SVM and Naïve Bayes classifiers	A.K.Jain et al [37]
PhishBlock	neural networks based SVM	M.A.Fahmy [24]
PhishMon	K Nearest Neighbors (KNN), AdaBoost, Random Forest (RF)	A.Niakanlahiji [50]
PILFER	Random Forest (RF)	I.Fette et al [26]
PhishStack	Random Forest (RF)	S.S.M Rahman et al [55]
MailTrout	Bidirectional Long Short-Term Memory (BLSTM) networks	P.Boyle et al [12]
SpamAssassin	Bayesian Additive Regression Trees (BART)	A.K Seewald et al [61]
Automated Individual Whitelist (AIWL)	Naïve Bayesian classifier	Y.Cao et al [14]
MMSPhiD	Machine learning approach (NNet), typosquatting-based approach, phoneme-based approach	G.Sonowal et al [65]
CBR-PDS	K nearest neighbours (kNN)	H.Abutair et al [3]

3 Methodology

To promote reproducibility of this paper, a detailed description of the test environment and algorithm implementation is presented.

In this paper, we explore the use of a deep learning method to remove the need for human interaction during the training phase. Specifically, the first contribution of this paper is the performance analysis of two popular bayesian-based software spam filtering solutions, and the novel application of a model based on *simple word embedding* and *global max pooling* referred to hereafter as (SWEM-*max*). The proposed method can more efficiently analyse topics within email messages since relationships between words are captured and the statistical structure of the language is mapped within a geometric space to improve accuracy. Previous research has shown this combination of techniques to outperform other forms of neural networks such *RNN*, *CNN* during training [64].

3.1 Data Sources

To evaluate the detection methods in this study the publicly available *Spam Assassin Dataset*[1] was used. The collection consisted of 6046 emails classified as either *ham:0* or *spam:1*. Following preprocessing the collection was reduced to 5293 (*ham:3915, spam:1378*) before being split using an 80:20 training/test ratio.

Before the dataset could be used we needed to address the imbalanced nature of the data. This was important because our proposed deep neural network would be comprised of multiple non-linear hidden layers, to form a sophisticated model able of learning very complex relationships. However, in most cases these relationships are likely to be a result of the sampling noise that exists in the training data, but not the test data. Therefore, even if a models test and training data are from the same dataset, there is a risk of overfitting the model. This risk is further increased when the imbalanced dataset is used for training. Since our dataset was imbalanced (74% ham to 26% spam) and would be used to train our deep learning model we first needed to select an appropriate technique to address this challenge. A number of such techniques exist in the literature, namely: reweight, weight sharing [57] and Deep synthetic minority oversampling technique (SMOTE) [18]. To avoid overfitting our model we selected to use a technique called Dropout [66]. In this method a unit (hidden and visible) of the neural network, along with its incoming and outgoing connections, is temporarily removed to prevent it from co-adapting too much. This dropout happens randomly during the training phase and can drastically reduce overfitting of the model. To increase accuracy and avoid overfitting the model we used multiple drop out layers as shown in Algorithm 1.

[1] https://www.kaggle.com/

3.2 Deep Learning Spam Detection Method (SWEM-max)

The second contribution of this paper is the application of *Deep Learning* to the problem domain to remove the need for human input during the training phase. To implement the model we first converted the dataset into a multidimensional dataset, using a technique called *simple word embedding*. Unlike other methods such as *one-hot encoding* where relationship information between words is missed, the words were represented as dense word vectors and the statistical structure of the language was mapped within a geometric space called an *embedding space*. Thus, the embedding layer formed the first layer within the neural network. Next, we down sampled the incoming feature vectors using *global max pooling* operation by only taking the maximum value of the time dimension. Models such as those presented in this paper can be prone to overfitting, therefore, the model was passed through a dropout layer where input units were frequently set to *0* at each step during the training phase to prevent overfitting. Next, to reduce the dimensionality of the data the model was passed through a number of dense layers. Here, the output layer was a dense layer with the dimension of *1* and the *sigmoid activation* function was used to narrow the output value between [0,1]. Finally, the formed neural network was compiled with *Adam optimiser* and *Binary Cross Entropy* loss function, and trained over 20 epochs (See Algorithm 1).

Algorithm 1 SWEM-max Spam Detection Algorithm

```
TrainAndValidate (training data, test data)
model ← sequential()
loss ← binary cross entropy
optimizer ← adam
epochs ← 20
Get input shape from training data
Add Embedding layer as the input layer
Add global max pooling operation for 1D layer
Add a drop out layer
Add new Dense Layer with relu activation
Add a drop out layer
Add Dense layer with sigmoid activation
Compile model using Optimiser and Loss
repeat
    /*Fit Model*/
    for i ← 1, epochs do
        Evaluate Loss
        Evaluate Validation Loss
        Evaluate Accuracy
        Evaluate Validation Accuracy
    end for
until All epochs completed
Return (Loss, ValLoss, Acc, ValAcc)
```

4 Results and Discussion

To test the performance of the two bayesian-based software software spam filters a local *email server*[1] was configured with default settings and training emails (*ham:3132*, *spam:1102*) sent from the dataset to the spam filters using the python *smtplib* module. Once received, each email was manually classified as either *ham/spam* to train the filters. Next, the same method was used to send the test emails (*ham:783*, *spam:276*) to each filter and performance was recorded.

To evaluate the performance of the methods in this study we measured the *tp*: true positive, *fp*: false positive, *tn*: true negative and *fn*: false negative rates and used these to calculate the classification accuracy as specified below.

1. **True positive (*tp*):** spam that is successfully detected
2. **False positive (*fp*):** ham email that is incorrectly classified as spam
3. **True Negative (*tn*):** ham email that is successfully classified as ham
4. **False Negative (*fn*):** spam email that is missed and classified as ham

While accuracy is a key performance indicator found within the literature other metrics derived from information retrieval and decision theory can help gain better insights into the obtained results [27]. Therefore, the following metrics were also calculated using the equations below where the Detection Rate *(DR)* signifies the ratio of spam instances detected by the model. The False Alarm Rate *FAR* signifies a ratio of misclassified email instances.

Detection Rate (DR): Defined as the % ratio of the number of true positive (*tp*) emails divided by the sum of true positive (*tp*) and false negative (*fn*) classified emails.

$$DR = \frac{tp}{tp+fn},\ \textit{detection rate} \in [0,1]$$

False Alarm Rate (FAR): Defined as the % ratio of the number of false positive (*tp*) emails divided by the sum of true negative (*tn*) and false positive (*fp*) classified emails.

$$FAR = \frac{fp}{tn+fp},\ \textit{false alarm rate} \in [0,1]$$

Precision (P): Defined as the % ratio of the number of true positive (*tp*) records divided by the sum of true positive (*tp*) and false positive (*fp*) classified records.

$$P = \frac{tp}{tp+fp},\ \textit{precision} \in [0,1]$$

Recall (R): Defined as the % ratio of number of true positive records divided by the sum of true positive and false negative (*fn*) classified records.

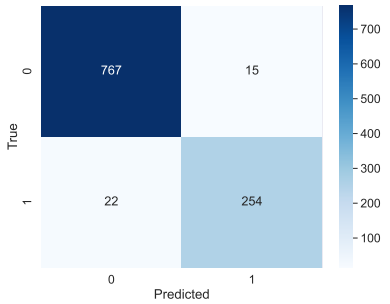
$$R = \frac{tp}{tp+fn},\ \textit{recall} \in [0,1]$$

F Measure (F_1): Defined as the harmonic mean of precision and recall and represents a balance between them. It is often used to measure the performance

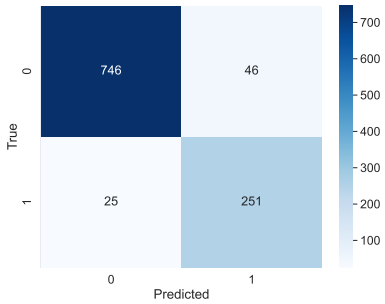
[1] https://www.hmailserver.com/

of a system when a single number is preferred [69].

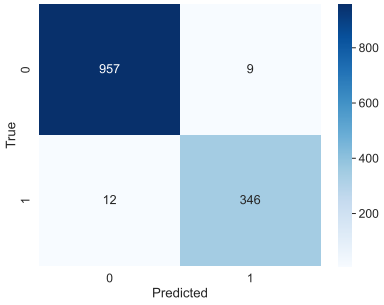
$$F_1 = 2 * \frac{P * R}{P+R},\ F\ Measure \in [0,1]$$



(**a**) Thunderbird



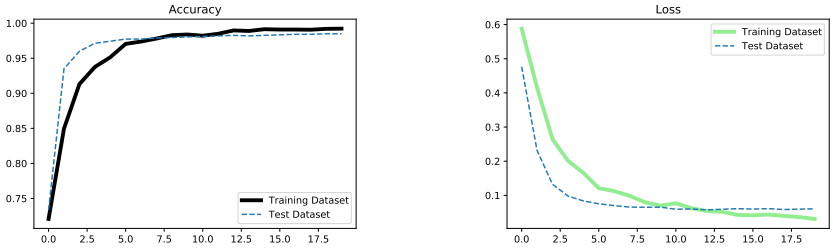
(**b**) Mailwasher



(**c**) Deep Learning

Fig. 1: Confusion Matrices for Detection of Spam Email

Fig. 1 shows the calculated confusion matrices based on the metrics described above. Fig. 2 shows the accuracy and loss values for the implemented SWEM-*max* model. Finally, Table 3 shows a performance comparison between the two bayesian-based software spam filtering solutions and the deep learning approach. It can be clearly seen that classification accuracy is improved using the Deep Learning approach (98.41%) when compared to both Mailwasher (93.29%) and Thunderbird (96.50%). In addition, *Precision*: (97.50%), *Recall*: (96.64%), *F1 Measure*: (97.05%), *Detection Rate*: (96.64%) and *False Alarm Rate*: (00.93%) were also shown to have improved. Importantly, the use of unsupervised machine learning removed the need for any human input during the training phase making it a more scalable and robust method.



(**a**) Accuracy (**b**) Loss

Fig. 2: Accuracy and Loss of SWEM-*max* model

Table 3: Comparison of detection methods based on Spam Assassin Dataset

Method	**Accuracy** *(a)*	**Precision** *(p)*	**Recall** *(r)*	**F1-Score** *(f_1)*	**Detection** *(dr)*	**False Alarm** *(far)*
Mailwasher	0.9329	0.8451	0.9094	0.8760	0.9094	0.0588
Thunderbird	0.9650	0.9442	0.9203	0.9321	0.9203	0.0192
SWEM-*max*	0.9841	0.9750	0.9664	0.9705	0.9664	0.0093

Before we can answer the question if the use of deep learning can remove the need for human input in spam email detection we must consider the role of the human in a wider context. As previously discussed, users are often considered to be the weakest link in many computer systems [13], especially in a human-centred environment[25]. Furthermore, many security incidents have been found to be caused by unintentional mistakes, or due to habitual behaviour that promotes an automatic response, rather than malicious acts by an attacker [72] In this respect, users' lack of understanding of how computer systems work, a lack of attention to security and the high quality visual deception deployed by phishers can weaken human defences [21]. From a spear phishing perspective, the human factor is especially inherent and can pose great danger to employees and organisations due to the inherent weakness of humans to identify every threat from spear phishing cues [51]. This can result in spam filtering software, such as those evaluated in this study, not being adequately trained. Additionally, this situation presupposes that the end user has taken time to train and retrain the filter in the first place, which may not be the case. Research on the human factor in cyber-enabled and cyber-dependent crime indicated that individuals' online behaviours facilitate cyber-dependent crime victimisation [4]. Hence, computer users put their organisations at risk of spam attacks through various social engineering tricks implemented by online criminals [63]. In this respect, the human factor is the underlying reason why many attacks are successful [58].

Effective information security education, training and awareness (SETA) program is essential for protecting organisational information resources, however, the increasing number of incidents resulting from employee noncompliance with security policy may indicate that many current SETA programs are not as effective nor optimal in changing employee behaviour to comply with security policy[8]. Although organisations provide cyber awareness training for their staff, attackers are able to bypass human defences in various ways such that even experienced staff make mistakes and can be deceived [11]. Since a secure system relies on humans making good decisions, human factors may create weaknesses in a system that an attacker could exploit [20]. In response, organisations often use rule-based training to teach individuals to identify threats in order to mitigate phishing's impact, however, even regular repetition of rule-based training may not yield increasing resistance to attacks [38].

Machine learning algorithms have been applied by researchers to automatically detect spam [15]. A bottleneck in developing such machine learning techniques is the lack of high quality labelled training data where human labelling to obtain high quality labelled data is expensive and not scalable [45]. Additionally, manually classifying spam can be time consuming. To be done effectively users are required to spend considerable time reading email messages and deciding whether it is spam or not. As such, some e-mail service providers prefer to automate spam detection using server based spam detectors and filters that can classify e-mails as spam automatically [53]. While normal spam filters has proven very useful by focusing on the content of email, they do not prevent the bandwidth from being wasted and is ineffective against the clever manipulation of the spam content by spammers [43]. Furthermore, spam filtering methods usually compare the contents of emails against specific keywords, which are not robust as the spammers frequently change the terms used in emails [71]. In this respect, automatic email filtering using machine learning may be the most effective method of detecting spam as spammers can easily bypass common spam filtering methods (text analysis, white and blacklisting of domain names, and community-primarily based techniques) easily [42].

The final area of consideration is the ever-changing landscape of spam email detection. As shown in section 2 the methods and techniques used by attackers are becoming increasingly more sophisticated and targeted. This was clearly demonstrated in [35, 68, 1] where the methods used by attackers during the current covid-19 pandemic became very specific, exploiting vulnerabilities in human psychology to leverage the fear surrounding the pandemic and generate phishing emails unique to this context. Here, it is unclear whether automated spam filtering, including the use of deep learning, can sufficiently understand the context in order to accurately detect spam email. A lack of available datasets based on this context prevented us from exploring this further, however, it was accepted that given the timely context, human input in the form of manual training would be advantageous. As such, the question of contextualisation in spam email detection remains unanswered.

5 Conclusions and Future Work

In this study, we evaluated the current state of spam detection and sought to explore the question: *Does the use of deep learning remove the need for human input in spam email detection?*. In doing so, we demonstrated an improvement in accuracy through the use of deep learning. The results in Section 4 showed that our deep learning model, based on simple word embedding and global max pooling (SWEM-max), returned higher detection accuracy (98.41%) than both the bayesian-based software spam filters: Thunderbird (95%) and Mailwasher(92%). However, while we demonstrated that replacing human input with machine learning during the training phase can improve accuracy, we postulated that this may not paint a full picture of the current state of affairs. We demonstrated that new attacks may target the same heuristics and cognitive biases (e.g. fear), but in new and unique ways, and in the context of a specific event. This was demonstrated in [59] where phishing attacks specifically targeted an increase in Zoom usage, and other remote working tools, due to the Covid-19 pandemic. As such, good research is often said to raise more questions than it answers. By that standard, this study raises new questions relating to the role of human input in spam detection. Future avenues of research could be motivated by follow-up questions such as:

1. What is the role and impact of context in spam detection?
2. Does using a dataset out of context affect detection performance?
3. Are new metrics beyond those used on section 4 needed for spam detection?

References

1. Abroshan, H., Devos, J., Poels, G., Laermans, E.: Phishing happens beyond technology: The effects of human behaviors and demographics on each step of a phishing process. IEEE Access **9**, 44928–44949 (2021)
2. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit. pp. 60–69 (2007)
3. Abutair, H., Belghith, A., AlAhmadi, S.: Cbr-pds: a case-based reasoning phishing detection system. Journal of Ambient Intelligence and Humanized Computing **10**(7), 2593–2606 (2019)
4. Akdemir, N., Lawless, C.J.: Exploring the human factor in cyber-enabled and cyber-dependent crime victimisation: A lifestyle routine activities approach. Internet Research (2020)
5. Alghoul, A., Al Ajrami, S., Al Jarousha, G., Harb, G., Abu-Naser, S.S.: Email classification using artificial neural network. International Journal of Academic Engineering Research (IJAER) **2**(11) (2018)
6. Alkahtani, H.S., Gardner-Stephen, P., Goodwin, R.: A taxonomy of email spam filters. In: Proceedings of the 12 th International Arab Conference on Information Technology (ACIT'11). pp. 351–356 (2011)
7. Alrwais, S., Yuan, K., Alowaisheq, E., Li, Z., Wang, X.: Understanding the dark side of domain parking. In: 23rd {USENIX} Security Symposium ({USENIX} Security 14). pp. 207–222 (2014)

8. Alshaikh, M., Naseer, H., Ahmad, A., Maynard, S.B.: Toward sustainable behaviour change: an approach for cyber security education training and awareness (2019)
9. Baadel, S., Lu, J.: Data analytics: intelligent anti-phishing techniques based on machine learning. Journal of Information & Knowledge Management **18**(01), 1950005 (2019)
10. Banday, M.T., Jan, T.R.: Effectiveness and limitations of statistical spam filters. arXiv preprint arXiv:0910.2540 (2009)
11. Bhardwaj, A., Sapra, V., Kumar, A., Kumar, N., Arthi, S.: Why is phishing still successful? Computer Fraud & Security **2020**(9), 15–19 (2020)
12. Boyle, P., Shepherd, L.A.: Mailtrout: a machine learning browser extension for detecting phishing emails. In: 33rd British Human Computer Interaction Conference: Post-Pandemic HCI–Living digitally. Association for Computing Machinery (ACM) (2021)
13. Caldwell, T.: Training–the weakest link. Computer Fraud & Security **2012**(9), 8–14 (2012)
14. Cao, Y., Han, W., Le, Y.: Anti-phishing based on automated individual white-list. In: Proceedings of the 4th ACM workshop on Digital identity management. pp. 51–60 (2008)
15. Chen, C., Zhang, J., Xie, Y., Xiang, Y., Zhou, W., Hassan, M.M., AlElaiwi, A., Alrubaian, M.: A performance evaluation of machine learning-based streaming spam tweets detection. IEEE Transactions on Computational social systems **2**(3), 65–76 (2015)
16. Christina, V., Karpagavalli, S., Suganya, G.: Email spam filtering using supervised machine learning techniques. International Journal on Computer Science and Engineering (IJCSE) **2**(09), 3126–3129 (2010)
17. Cveticanin, N.: https://dataprot.net/ (2021)
18. Dablain, D., Krawczyk, B., Chawla, N.V.: Deepsmote: Fusing deep learning and smote for imbalanced data. IEEE Transactions on Neural Networks and Learning Systems (2022)
19. Dada, E.G., Bassi, J.S., Chiroma, H., Adetunmbi, A.O., Ajibuwa, O.E., et al.: Machine learning for email spam filtering: review, approaches and open research problems. Heliyon **5**(6), e01802 (2019)
20. Desolda, G., Ferro, L.S., Marrella, A., Catarci, T., Costabile, M.F.: Human factors in phishing attacks: A systematic literature review. ACM Computing Surveys (CSUR) **54**(8), 1–35 (2021)
21. Dhamija, R., Tygar, D.: Hearst. m. 2006. why phishing works. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 22–27
22. Dhanaraj, S., Karthikeyani, V.: A study on e-mail image spam filtering techniques. In: 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering. pp. 49–55. IEEE (2013)
23. Dudley, J.: Improving the performance of heuristic spam detection using a multiobjective genetic algorithm. School of Computer Science and Software Engineering, The University of Western Australia (2007)
24. Fahmy, H.M., Ghoneim, S.A.: Phishblock: A hybrid anti-phishing tool. In: 2011 International Conference on Communications, Computing and Control Applications (CCCA). pp. 1–5. IEEE (2011)
25. Fan, W., Kevin, L., Rong, R.: Social engineering: Ie based model of human weakness for attack and defense investigations. IJ Computer Network and Information Security **9**(1), 1–11 (2017)

26. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th international conference on World Wide Web. pp. 649–656 (2007)
27. Gangavarapu, T., J.C..C.B.: Applicability of machine learning in spam and phishing email filtering: review and approaches. Artificial Intelligence Review **53**, 5019–5081 (2020)
28. Guo, K.H., Yuan, Y., Archer, N.P., Connelly, C.E.: Understanding nonmalicious security violations in the workplace: A composite behavior model. Journal of management information systems **28**(2), 203–236 (2011)
29. Gupta, B.B., Arachchilage, N.A., Psannis, K.E.: Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommunication Systems **67**(2), 247–267 (2018)
30. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2011)
31. Hao, S., Syed, N.A., Feamster, N., Gray, A.G., Krasser, S.: Detecting spammers with snare: Spatio-temporal network-level automatic reputation engine. In: USENIX security symposium. vol. 9 (2009)
32. Harinahalli Lokesh, G., BoreGowda, G.: Phishing website detection based on effective machine learning approach. Journal of Cyber Security Technology **5**(1), 1–14 (2021)
33. Hayati, P., Potdar, V., Talevski, A., Firoozeh, N., Sarencheh, S., Yeganeh, E.: Definition of spam 2.0: New spamming boom. pp. 580 – 584 (05 2010). https://doi.org/10.1109/DEST.2010.5610590
34. Heron, S.: Technologies for spam detection. Network Security **2009**(1), 11–15 (2009)
35. Hill, J.: https://abnormalsecurity.com/blog/how-to-stop-email-spoofing (2021)
36. Irwin, L.: https://www.itgovernance.eu/blog/en/the-5-most-common-types-of-phishing-attack (2020)
37. Jain, A.K., Gupta, B.: Phish-safe: Url features-based phishing detection system using machine learning. In: Cyber Security, pp. 467–474. Springer (2018)
38. Jensen, M.L., Dinger, M., Wright, R.T., Thatcher, J.B.: Training to mitigate phishing attacks using mindfulness techniques. Journal of Management Information Systems **34**(2), 597–626 (2017)
39. Khan, S.A., Khan, W., Hussain, A.: Phishing attacks and websites classification using machine learning and multiple datasets (a comparative analysis). In: International Conference on Intelligent Computing. pp. 301–313. Springer (2020)
40. Kołcz, A., Chowdhury, A.: Hardening fingerprinting by context. CEAS'07 (2007)
41. Kołcz, A., Chowdhury, A., Alspector, J.: The impact of feature selection on signature-driven spam detection. In: Proceedings of the 1st Conference on Email and Anti-Spam (CEAS-2004) (2004)
42. Kumar, N., Sonowal, S., et al.: Email spam detection using machine learning algorithms. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). pp. 108–113. IEEE (2020)
43. Lam, H.Y., Yeung, D.Y.: A learning approach to spam detection based on social networks. Ph.D. thesis, Hong Kong University of Science and Technology (2007)
44. Liu, X., Zou, P., Zhang, W., Zhou, J., Dai, C., Wang, F., Zhang, X.: Cpsfs: A credible personalized spam filtering scheme by crowdsourcing. Wireless Communications and Mobile Computing **2017** (2017)
45. Luo, C., Xia, C., Shao, H.: Training high quality spam-detection models using weak labels (2020)
46. Mansoor, R., Jayasinghe, N.D., Muslam, M.M.A.: A comprehensive review on email spam classification using machine learning algorithms. In: 2021 International Conference on Information Networking (ICOIN). pp. 327–332. IEEE (2021)

47. Mohamed, S.A.E.: Efficient spam filtering system based on smart cooperative subjective and objective methods (2013)
48. Mozilla Support: Thunderbird and junk / spam messages (2021), https://support.mozilla.org/en-US/kb/thunderbird-and-junk-spam-messages, [Accessed on 2021-10-19]
49. Najadat, H., Hmeidi, I.: Web spam detection using machine learningin specific domain features (2008)
50. Niakanlahiji, A., Chu, B.T., Al-Shaer, E.: Phishmon: A machine learning framework for detecting phishing webpages. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 220–225. IEEE (2018)
51. Nicho, M., Fakhry, H., Egbue, U.: Evaluating user vulnerabilities vs phisher skills in spear phishing. Internat J Comput Sci Inform Syst **13**, 93–108 (2018)
52. Paans, R., Herschberg, I.: Computer security: the long road ahead. Computers & Security **6**(5), 403–416 (1987)
53. Patidar, V., Singh, D., Singh, A.: A novel technique of email classification for spam detection. International Journal of Applied Information Systems **5**(10), 15–19 (2013)
54. Patil, R.C., Patil, D.: Web spam detection using svm classifier. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO). pp. 1–4. IEEE (2015)
55. Rahman, S.S.M.M., Islam, T., Jabiullah, M.I.: Phishstack: evaluation of stacked generalization in phishing urls detection. Procedia Computer Science **167**, 2410–2418 (2020)
56. Ramachandran, A., Dagon, D., Feamster, N.: Can dns-based blacklists keep up with bots? In: CEAS (2006)
57. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: International conference on machine learning. pp. 4334–4343. PMLR (2018)
58. Richardson, M.D., Lemoine, P.A., Stephens, W.E., Waller, R.E.: Planning for cyber security in schools: The human factor. Educational Planning **27**(2), 23–39 (2020)
59. Roman, R., Zhou, J., Lopez, J.: An anti-spam scheme using pre-challenges. Computer Communications **29**(15), 2739–2749 (2006)
60. Rosenthal, M.: https://www.tessian.com/blog/phishing-statistics-2020/ (2021)
61. Seewald, A.K.: Combining bayesian and rule score learning: Automated tuning for spamassassin. Intelligent Data Analysis. Technical report, TR-2004-11 Austrian Research Institute for Artificial Intelligence, Vienna, Austria (2004)
62. Sendpulse: https://sendpulse.com/support/glossary/phishing (2020)
63. Shakela, V., Jazri, H.: Assessment of spear phishing user experience and awareness: an evaluation framework model of spear phishing exposure level (spel) in the namibian financial industry. In: 2019 international conference on advances in big data, computing and data communication systems (icABCD). pp. 1–5. IEEE (2019)
64. Shen, D., Wang, G., Wang, W., Renqiang, M., Su, Q., Zhang, Y., Li, C., Henao, R., Carin, L.: Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms (05 2018). https://doi.org/10.18653/v1/P18-1041
65. Sonowal, G., Kuppusamy, K.: Mmsphid: a phoneme based phishing verification model for persons with visual impairments. Information & Computer Security (2018)
66. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)

67. Statista: https://www.statista.com/statistics/255080/number-of-e-mail-users-worldwide (2021)
68. Tessian: https://www.tessian.com/blog/covid-19-real-life-examples-of-opportunistic-phishing-emails-2/ (2021)
69. Tong, Z., Weiss, S.M.: The handbook of data mining. Lawrence Erlbaum Assoociates (2003)
70. Wash, R., Cooper, M.M.: Who provides phishing training? facts, stories, and people like me. In: Proceedings of the 2018 chi conference on human factors in computing systems. pp. 1–12 (2018)
71. Wu, C.H.: Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. Expert systems with Applications **36**(3), 4321–4330 (2009)
72. Zafar, H., Randolph, A., Gupta, S., Hollingsworth, C.: Traditional seta no more: investigating the intersection between cybersecurity and cognitive neuroscience. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)