20210013

[Email address]

Atrificial Intellegence and machine learning coursework

Detecting phishing and spam emails

Contents

[*Introduction* 2](#_Toc103803745)

[*Background Research* 4](#_Toc103803746)

[*Design and Methodology* 7](#_Toc103803747)

[*Results* 10](#_Toc103803748)

[*Future Work* 11](#_Toc103803749)

[*References* 12](#_Toc103803750)

# *Introduction*

In this paper, I will be exploring the methodology behind how one can create an Artificial Intelligence and Machine learning algorithm that can detects phishing and spam emails, how you can make it more efficient and attempt to create a Machine learning algorithm to solve this problem.

What are spam emails? Spam emails are unwanted junk emails that are sent out in bult by botnets, network of infected computers, in discriminately. Spam emails are mostly sent for commercial purposes as the cost per email is incredibly low however spam emails could also have malicious attempt to gain access to your computer and data.

Phishing is an illegal practise where people or companies attempt to imitate emails and attempt to deceive the victim into giving them something they want; this could be money or personal data. Phishing remains one of the most fraudulent tricks found in email traffic. Usually, this phishing emails are disguised as big, well known, companies and they take you to a fake website that looks similar to the big companies and ask you to enter sensitive data to it.

Phishing is a real big problem in the modern world. It can ruin peoples lives and can affect even the biggest of companies, in 2019, Kaspersky’s anti-phishing system identified [467 million attempted transitions](https://securelist.com/spam-report-2019/96527/) to phishing websites. Nearly one-in-seven of our users has faced this threat. (Benkovich, 2022)

Phishing techniques have been evolving rapidly over time, however, so has the detecting methods.

So why use AI and machine learning to detect phishing emails? Well, machine learning algorithms can learn from historical fraud patterns and use the same patterns to detect future potential fraud transactions. Machine learning algorithms are far more effective than humans at detecting fraudulent emails and are faster at processing information. They are also capable of detecting sophisticated fraud traits that a normal person simply cannot pick up on.

Machine learning algorithms work faster than humans, are more efficient that humans and they scale better. As Machine learning algorithms can take over the repetitive work of manual fraud analysis and can take over routine tasks. Also, machine learning algorithms show better performance as the size of the data set increases. So, the more the algorithm is used on phishing emails, the better it will detect and recognise fraudulent emails.

There are many different types of fraud that an algorithm like this can detect, for example, email phishing, payment fraud, identity theft, but I will mostly be focusing on email phishing.

# *Background Research*

How does artificial intelligence detect phishing emails?

Look for anomalies and warning signals:

This means the artificial intelligence will look at the email behaviour and the intent and content of the email. If the emails show signs of urgency, this is a big sign that the email may potentially be a phishing scam. The artificial intelligence will detect this, and the warning signal will be lit. The artificial intelligence will also look at the context of the content within the email. This can help the artificial intelligence understand the email more precisely whether the email is a threat or not. It also allows for the artificial intelligence to show better distinction between phrases as it will be able to differentiate between normal spam and phishing spam. The artificial intelligence also takes into account the email header and will analyse it to check for email spoofing, misspelled domains and other types of spoofing.

Analysing the context of the message:

One of the main elements of phishing emails is that they evoke human emotions and concerns. For example, a person waiting for an important parcel will more likely click on a link in a phishing email and fill in the proposed fields on that website. In combination with this, the phasing email may make use of emotive language to manipulate the recipient into downloading an attachment or clicking a link. The artificial intelligence will analyse the whole message instead of just comparing it to previous phishing emails. Graphical user interface, text, application

Description automatically generated

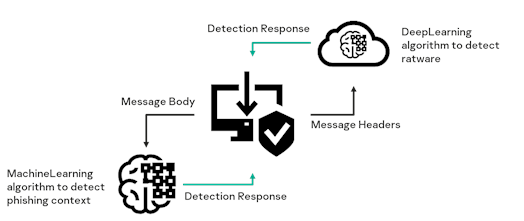
Graphical user interface, text, application, email

Description automatically generated

Examples of phishing emails: (Benkovich, 2022)

The algorithm will take into account, previous interaction between the sender and recipient, any data that may be requested by the sender and the topic of the subject itself and with the addition of machine learning, the algorithm can constantly improve, making analysis even better and more accurate. It will also look out for common phishing phrases.

Here is an example of how an email can be processed by an algorithm



(Benkovich, 2022)

# *Design and Methodology*

My machine learning algorithm will use a supervised learning approach to detect phishing and spam emails.

To create my machine learning algorithm, I used Google Colab as it allows everyone to write and execute python code online, without the need for installing software and is well suited for machine learning algorithms. Google Colab is hosted by Jupyter notebook, so this allows everybody access to its computing resources including GPUs, so you don’t have to run algorithms on your machine but instead use resources offered online for free.

To get my dataset, I used Kaggle, which is a subsidery of Google and is an online community of data scientists and machine learning practitioners. This website has tons of resources for AI and ML algorithms, such as data sets, where I was able to find mine. I used a data set called Email Spam Dataset which was created by Nitisha. This dataset contains

Firstly, I identified which libraries I would need to create my machine learning algorithm. I decided that I would need NumPy, Pandas and Natural Language Tool Kit.

I would need pandas as it would be crucial for my algorithm. Pandas allows for manipulation and analysis of data. It offers data structures and operations that I would need to manipulate the dataset.

I would need Natural Language Tool Kit (NLTK) for manipulating the content of the dataset to make it more readable and to make the algorithm run more efficiently on the dataset.

Firstly, I imported the libraries that I mentioned before into Google colab. I then read the dataset file and stored it in a variable called df. To test that the test file was read correctly, I printed the first 5 rows of the text file.

I then checked the shape of the dataset, which mean I would be checking the number of rows and columns in the dataset. To do this I made use of pandas function called shape, which returns the number of rows and columns.

I then used .columns to get the names of the of columns in the data set.

Then I preceded to check whether there are any duplicating data within the dataset. Pandas has a function which allows you to drop duplicate rows of data.

I checked the shape of the data set again after calling the drop\_duplicates to check whether there was any duplicate data in the data set.

To make sure there were no empty/null field in the dataset, I called isnull function and found that there was 1 empty field. To remove this row of data from the data set I made use of the dropna function which drops any row that contains an empty field. I then called isnull again to verify that the empty/null row was removed.

Now that the data set has been cleaned up, I began tokenizing the dataset, this means I started to format the dataset so that it removes all unnecessary information from the contents of the dataset and splitting the text into minimal meaningful units.

To do this I made use of NLTKs stopwords. Stop words are words such as “the”, “and” “I”, etc, so by using the NLTK stopwords, I can filter out all the common words in the dataset so that we are left with only unique words.

I created a function that went through the whole data set and removed all the punctuation from the data set so that only words were left. I then removed all the stop words in the data set and then finally returned the data set that only contains unique words.

I converted the dataset collection into a list of tokens where each cell of the Body column is a token.

I then split the data set for training and testing. I split the data set 80:20, so 80% is used for training and 20% is used for testing.

To make my predictions I used Naïve Bayes Classifier model. I decided to use Naïve Bayes Classifier model as it is fast to predict the class of the data set and it is ideal for large, multi-class, datasets, and you can use it with less training data. It is also commonly used in text classification/spam filtering making it ideal for my algorithm.

I call sklearn’s multinomialNB function and pass in the training data.

The multinomialNB learning approach is popular in natural language processing.

# *Results*

When evaluating the model on the training data. My algorithm came out with a 0.964 accuracy and when evaluating the model on the test data, the algorithms accuracy dropped very slightly to a 0.956. This shows that my supervised algorithm is very accurate but this also shows that some improvements could be made to make the algorithm more accurate.

# *Future Work*

To build upon this algorithm to make it more efficient and smarter, in the future I could make this algorithm an unsupervised algorithm instead of a supervised algorithm. This would mean that the algorithm would have to identify and learn all the patterns in the fields, rows and columns by itself. This would make the algorithm ideal for anomaly detection, which is very useful for detecting fraudulent transactions, such as spam emails and phishing. This would also allow the algorithm to be come more adaptive as the more the algorithm is run, the better it would perform.

# *References*

Services, P. and Security, E., 2022. *What Is Spam Email?*. [online] Cisco. Available at: <https://www.cisco.com/c/en/us/products/security/email-security/what-is-spam.html> [Accessed 7 May 2022].

En.wikipedia.org. 2022. *Email spam - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Email\_spam> [Accessed 7 May 2022].

Gatefy. 2022. *How artificial intelligence and machine learning fight phishing - Gatefy*. [online] Available at: <https://gatefy.com/blog/how-ai-and-ml-fight-phishing/> [Accessed 7 May 2022].

Benkovich, N., 2022. [online] Available at: <https://www.infosecurity-magazine.com/blogs/machine-learning-email-phishing/> [Accessed 7 May 2022].

Technology partner for innovative companies. 2022. *How to Use AI and Machine Learning in Fraud Detection*. [online] Available at: <https://spd.group/machine-learning/fraud-detection-with-machine-learning/> [Accessed 7 May 2022].

Malware Patrol. 2022. *AI-Enabled Phishing Tactics and Security Defenses - Malware Patrol*. [online] Available at: <https://www.malwarepatrol.net/ai-enabled-phishing/> [Accessed 7 May 2022].

Sindar Sinaga, A., 2021. *ShieldSquare Captcha*. [online] Iopscience.iop.org. Available at: <https://iopscience.iop.org/article/10.1088/1757-899X/1088/1/012011/pdf> [Accessed 7 May 2022].

Darktrace.com. 2022. *Comparing different AI approaches to email security*. [online] Available at: <https://www.darktrace.com/en/blog/comparing-different-ai-approaches-to-email-security/> [Accessed 7 May 2022].

Research.google.com. 2022. *Google Colab*. [online] Available at: <https://research.google.com/colaboratory/faq.html#:~:text=Colab%20allows%20anybody%20to%20write,learning%2C%20data%20analysis%20and%20education.> [Accessed 7 May 2022].

En.wikipedia.org. 2022. *Kaggle - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Kaggle> [Accessed 7 May 2022].