National College of Ireland

Human-centric Approach to Emails Phishing Detection

MSc Research Project
Data Analytics

Ermesa Pepe
Student ID: X20212887

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ermesa Pepe……

Student ID: X20212887……..……

Programme: Master in Data Analytics…… **Year:** 2022……..

Module: Research Project……

Supervisor: Vladimir Milosavljevic……

Submission Due Date: 15/08/22……

Project Title: Human-centric Approach to Emails Phishing Detection……

Word Count: ……10002…… **Page Count**……25……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: … *Ermesa Pepe* ……

Date: 15/08/2022……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Human-centric Approach to Emails Phishing Detection

Ermesa Pepe
X20212887

Abstract

In recent years, cyber-phishing attacks have become increasingly sophisticated, targeted, and tailored to be effective only when specific information is provided. It is, therefore, impossible for automated detection systems to be 100% accurate all the time, which increases the uncertainty surrounding expected behaviour when confronting a potential phishing email. Human-centric defence approaches, on the other hand, emphasise extensive user training. However, users are often unaware of the continuously emerging patterns. Identifying an email recipient's intentions and emotional triggers by analysing the email's text and summarising the most relevant parts is a fascinating approach to identifying these threats. As a solution, a transformer-based approach is proposed to analyse psychosocial triggers in emails, detect spam emails and possible malicious intent, and summarise emails. Using this information, we facilitate the user's ability to determine whether an email is phishing and learn advanced malicious patterns on their own.

1 Introduction

Emails represent one of the tools most adopted in communication worldwide. Unfortunately, due to its widespread use, email is subject to attacks such as SPAM, phishing, and bot infections. SPAM is essentially junk mail or unsolicited correspondence. At the same time, phishing is an evil technique that uses messages with links to copies of original sites to steal information from unsuspecting users.

Our daily lives are more affected by electronic mail than ever before. The growth of the internet has made electronic mail an essential tool for communication and work, offering low-cost instant messaging. Emails have become the second most popular form of communication after voice since people can now exchange information asynchronously. However, electronic mail has become a tedious and even risky task due to the volume of incoming messages (some of which are extremely important, but others are spam).

Yet, emails are among the most popular online activities in the actual hybrid workplace and remain a major tool for communication and collaboration. However, due to the current regulation and the risk of loss of money and reputation that cannot be estimated, companies spend large amounts of time and money implementing software to protect their information, with an average online security budget of around 10%. (Frank Cremer, 2019)

However, *"human hardware"* is the most vulnerable element, and companies should operate on a preventive rather than a cure basis. Human error is the cause of at least 95% of

cybersecurity breaches (Technology, 2014); with designed tools and awareness training courses, this number can be drastically reduced. The employee is the last line of defence in a company's security, the *"human firewall"* from SMEs to large enterprises. In 2021 in mid-April, Google's Threat Analysis Group reported blocking 18 million COVID-19-themed malware and phishing emails daily. (Huntley, 2021)

Phishing attacks are the top cause of cybersecurity breaches like ransomware infections; research suggests that 91% of successful cyber-attacks result from phishing scams. (Research, 2022)

Even if companies are increasing security mechanisms, phishing is still a growing threat in 2022, partly due to a lack of awareness among employees. *Spear Phishing* is the most sophisticated and targeted attack that targets specific company workers. (Irwin, 2022)

For example, an email impersonating the CEO is likely to be clicked by most employees and may contain malware as an attachment. These attackers' success has inspired innovative and advanced developments. This threat can be dramatically reduced with innovative tools to support employees in recognising potentially harmful emails and reporting suspicious ones. The risk can be reduced, in fact, by simulating customised phishing attacks and offering an analysis of the email, such as a summary of its content, its final intent, and the hidden cognitive/emotional threat. Although this is possible with machine learning techniques such as text summarisation, automatic spam detection, and intent detection applied to the context of electronic mail to extensively save time and retrieve precise email content meaning and sender purpose.

To guide the project, the research question has been structured as follows:

RQ. *"How can machine learning assist businesses in preventing ransomware attacks, increasing the employee's ability to detect phishing emails?"*

> ***Sub-RQ.*** *"How can a human-centric approach based on Transformers and Transfer learning outperform existing deep learning techniques in detecting potential phishing emails in the workplace?"*

The key objective of this research is to determine the effectiveness of applying a human-centric and context-aware approach in the contest of Natural language Processing (NLP), using transfer learning techniques by adopting lightweight Transformer-based models.

In a business context, the proposed solution wants to locate a wider *Cyber Security Awareness program* capable of reaching all employees. While existing solutions focus, in fact, on analysing the mitten email address, hyperlinks, and attachments, the shift is now on the email content analysis that, in many circumstances, still eludes the employee.

The latest research has changed its attention from automated phishing detection to helping people make decisions. (Zainab Alkhalil, 2021)

Real-time support, such as notifications, can help prevent risky behaviours since automatic filtering approaches are not always 100% accurate. In addition, providing users with security indicators' information allows them to capture unusual behaviour.

2 Related Work

A human-centric solution using autonomous ML agents to aid human judgment can be a crucial step in the right direction and requires a proper investigation of the most recent literature based on the latest machine learning techniques.
For the project scope, it is required to build different models to extract a variety of qualitative information from emails body and support employees' judgemental ability in case of suspicious emails. Therefore, a comprehensive current literature review is essential to building a human-centric model based on the fusion of different NPL techniques in fields such as text summarisation, intent detection, emotion detection and SPAM detection.
The following sections will review various learning techniques supporting our hypothesis.

2.1 Spam Detection

In recent years, Text Categorisation has proved to be a particularly active and constantly evolving research area. Many existing approaches are applied to the multitude of documents made available through the internet. In addition, Text Categorisation methods have been widely used in spam filtering problems.

Starting with Lewis et al. (1997) (Mantrach, 2015) is clear that email conversations between two or more people can be treated as a language processing task, giving effective results using textual matching methods of Information Retrieval (IR). For example, (Drucker, et al., 1999) analyse the use of extract Term Frequency - Inverse Document Frequency (TF-IDF) with Support Vector Machines to identify emails as spam or as not spam. The authors also compare the performance of other algorithms, including Ripper, Rocchio and boosted decision trees; SVMs give the best results in terms of accuracy and efficiency. (Hazarika, 2016)
The most efficient traditional model to filter spam and phishing emails leverage (Wagner, 2019) proposing a Random Forest classifier trained with features from "phishy" keywords and *URL* reputation scores to discover the most unseen phishing attacks. The technique collects reputation scores for URLs and determines the similarity scores of receivers using historical data from a sizable dataset.

Since attacks are evolving, using sophisticated techniques to elude filtering systems also researchers started to focus on different approaches, including the use of neuronal networks, Recurrent Neural networks (RNN) (Lukas Halgas, 2019) and Convolutional Neural networks (CNN) (Nikita Benkovich, 2020). In (Hussain, 2017), a CNN-based model was released containing features only from email headers to identify probable spam communications. Recent experiments demonstrate that combined elements from the body improve the detection performance against various phishing attacks. (Stamp, 2022)
The attention is shifted to analysing the most successful recent attacks and overcoming the limitation of the existing solutions. For example, most recent effective attacks don't contain malicious attachments or phishing URLs; however, they encourage victims to engage with the requested behaviour leveraging on the emotional/ psychological response of the victim as part of attack playbooks. (Zainab Alkhalil, 2021)

These limitations are well highlighted in (Theodore Longtchi, 2022) and support our model choice—which requires further investigation in text summarisation, intent detection and emotional recognition.

2.2 Text Summarisation

Radev et al. (2002) define a summary as "a text usually less than half of the original text and conveys important information in the original text".(Radev, 2002)

Due to the inability of people to assimilate vast amounts of information, automatic and efficient methods of summarising text are important with the explosion of data available in the form of unstructured text such as emails. (Patil, 2014)

Automatic Text Summarisation(ATS) should shorten the volume of information by creating a summary of sentences without losing any main content. Text Summarisation tasks can follow various approaches, such as extractive, abstractive or hybrid techniques to identify the most relevant sentences and remove irrelevant ones. (Adhika Pramita Widyassari, 2022)

Abstractive summarisation is one of the most challenging ATS tasks and requires a deep analysis of the input data. (K.Mohameda, 2021)

In the 1990s, Natural Language Processing (NLP) techniques came to a new stage; following cognitive psychology and linguistics analysis developments, methods for automatic text summarisation were developed based on machine learning as a classification problem (Rong Xu, 2006). As a result, deep learning has gained significant traction in text summarisation in recent years. *SummaRuNNer* is a Recurrent Neural Network (RNN) which implements a two-layer bidirectional GRU-RNN to perform extractive summarisation (Nallapati, 2016) (Jingli Shi, 2022). *SummaRuNNer* is one of the earliest neural approaches to adopt an encoder based on Recurrent Neural Networks, which leverage *REFRESH* (Afonso Mendes, 2019), a reinforcement learning-based system trained by globally optimising the *ROUGE* metric.

More recent work achieves higher performance with more sophisticated model structures.

With the most relevant work by (Xuanyi Dong, 2018), the adoption of neural networks has come later with little improvement on automatic metrics like ROUGE due to the complexity of the task.

In recent years, much research has been conducted to build efficient Pretrained Language Models (PLM) to solve NLP tasks due to the high computational resources consumed. For example, among various PLMs, *BERT* (Ashish Vaswani, 2017) has achieved SOTA results in various NLP tasks. (Liu, 2019)

One of the major limitations of BERT is the large model size which leads to difficulty pre-training the model from scratch. Hence, the advent of several compressed, optimised variants of *BERT*, like *DistilBERT* (Victor Sanh, 2020) and *ALBERT* (Zhenzhong Lan, 2020), to solve NLP tasks using *Transfer Learning*. (Miller, 2019)

Particularly efficient *BART* (Mike Lewis, 2020) outperforms *RoBERTa* on long articles using hybrid embeddings on the benchmark datasets compiled from open-domain articles, such as *CNN/DailyMail*. (Anon., 2020)

Moreover, Wang et al. explore the domain shift phenomenon and prove that a model trained on one domain dataset performs poorly on another different domain. (Danqing Wang, 2019)

The recent approach focuses on domain-specific text summarisation (Jiang & Wang, n.d.).
It can successfully summarise email To-Do items to help people overview emails and schedule daily work (Zhang, 2022). Because emails can be overwhelmingly long, lightweight BERT-like models are the most effective for the project scope. In this way, automatic text summarisation would allow users to examine the email's primary purpose without superfluous or distracting information clouding their judgement. For instance, an ideal summarisation pipeline would immediately point out if a part of the email evokes a potential Cyberthreat.

2.3 Intent Recognition

Studying users' intent can help design and develop an intelligent email assistant in phishing detection by increasing accessibility and visibility of the email scope.
Dabbish and Kraut found that users wish to email information to be more available at the surface level, confirming Whittaker's previous findings that users prefer availability and visibility. (Kraut, 2006)
Furthermore, Dabbish and Kraut found that users tend to have a small number of folders. A behaviour that increased the surface level visibility of individual email messages and reduced the feelings of email overload (Vidal, 2018) . Therefore, we conclude that the best solution:

- It should not increase complexity;
- It should instead make it easier for the user to retrieve the email's information.

In 2000 Kiritchenko et al. were the first to use SVMs (Matwin, 2000) to deal with the problem linked to the temporal flow of email. The proposed solution considers temporal relationships with the information from content-based learning methods. Martin et al. (Sewani, 2005) analysed the inbox traffic to identify the behavioural features useful for recognising spam. Finally, Bekkerman et al. (Bekkerman, 2004) performed a series of tests using famous benchmarks such as *Enron* corpora, comparing techniques such as *NB*, *SVM*, *MaxEntropy* and *Wide-Margin Winnow*, showing that SVM has better performance in almost all tests.

When dealing with emails, it is also important to consider the structure and fields. By reducing its feature space, we can decide which parts are useful for classification and which can be ignored. For example, Lampert et al. (Lampert, 2010) proposed a system that identified nine graphic, linguistic, and spelling segments within a message. A reduction to two segments produced encouraging results (accuracy of 93.6%).
Email flow may be extremely high in large contexts, such as large companies. Therefore, a non-incremental classification is not possible. Carmona et al. (Carmona-Cejudo, et al., 2011) present *GNUsmail*, an open-source framework for classifying real-time emails. *OzaBag*, *OzaBoost*, *Adaptive Hoeffding Trees*, *Majority Classes* and *DDM* are presented in conjunction with incremental NB on the *Enron* dataset. Whittaker and Sidner found that users often treat their inbox as an external memory store, with messages in view serving as reminders. (Matt Balogh, 2022)
Carden et al. (Lila Carden, 2021) also highlight the importance of task management in email.
Intent recognition problems have been addressed by researchers using a variety of deep learning techniques, such as Convolution Neural Networks (CNN), used to recognise user queries by extracting features in the form of vectors (Hashemi, 2016), and Recurrent Neural

Networks (RNN). In some cases, the RNN method yields good results when applied to the intent recognition problem. (Bhargava, et al., 2013)
An extension of *Long Short-term Memory* (LSTM), known as a *Gated Recurrent Unit* (GRU), can recognise more ordered words and is a less complex model. (Dey & Salem, 2017)
In the field of Intent Recognition(IR), Language model pre-training has an advanced state of the art as in other NLP tasks ranging from sentiment analysis to question answering, natural language inference, named entity recognition, and textual similarity. State-of-the-art pre-trained models include ELMo (Matthew E. Peters, 2019), *GPT* (Narasimhan, 2018), and, more recently, Bidirectional Encoder Representations from Transformers (BERT) (Jacob Devlin, 2019). For example, BERT combines word and sentence representations in a single very large Transformer helping to build a human-centric intelligent system for Intent recognition (BERT-IR). (Vasima Khan1, 2021) BERT is pre-trained on vast amounts of text, with an unsupervised objective of masked language modelling and next-sentence prediction and has the advantage that it can be fine-tuned with various task-specific goals.

2.4 Emotion Recognition

In the book *Thinking Fast and Slow*, Daniel Kahneman describes two separate systems of thought - the first, the emotional and intuitive process, and the second, slower rational logic. Emotional thoughts are incredibly powerful; they are quick to act, easy to follow and often provide an immediate reward. On the other hand, rational thoughts take time and effort to commit and often lead us to make better more strategic decisions. (Ala Mughaid, 2022)
Cybercriminals know this well, so they actively set out to trigger emotional responses, bypassing rational thoughts and thus increasing the likelihood of clicking on a link. For this reason, Emotion recognition (ER) is one of the most important and challenging studies contributing to phishing prevention. It is critical to recognise and monitor human emotions in live environments with a high accuracy rate and rapid recognition time to keep users informed of their triggered emotional responses. (Attacks, 2017)
In the field of phishing prevention, emotion recognition is largely investigated today. For example, research supports the hypothesis that older adults may be more susceptible to phishing than young adults. According to Zebrowitz et al. (2013), perceived trust increases as individuals age, whereas sensitivity to untruthful information decreases (Castle et al. 2012); Compared to young adults, older adults are more likely to invest in trust games generously (e.g., more generous investments).
Adversaries often exploit these behaviours to get individuals to act against their interests. Behavioural patterns are triggered by psychological principles of persuasion, which Cialdini identified as psychological principles of persuasion. According to him, there are six principles of persuasion: *authority*, *commitment*, *liking*, *reciprocity*, *scarcity*, and *social proof* (Hadnagy 2010). Several prior works have investigated the extent to which Cialdini's principles of persuasion (PoP) are used in phishing emails (Ferreira, 2019) (Attacks, 2017)

3 Research Methodology

It should be noted that the automatic classification of spam/phishing emails is an area of research that is not yet established and consolidated. The main reason is also the absence of a standard public dataset of emails to evaluate various classification methods and compare the results deriving from the work of multiple researchers. However, a public body of emails made available by Enron Corporation is available, originally consisting of 500,000 emails from the accounts of 150 people.

The *Enron*[1] *Email Dataset* is a large corpus of real-world emails subpoenaed from Enron Corporation that was placed in the public record and made available to researchers. Several dataset variants are labelled for different tasks, including sentimental analysis, speech acts, and text summarisation. The original dataset consists of over 500,000 email messages from the email accounts of 150 people. As a pre-processing step, we read each email and decompose it into its fields (From, To, Subject, Content, etc.). As highlighted in the literature review, despite the significant improvements resulting from deep sequence-to-sequence models and their mechanisms in NLP tasks, RNN-based models are still limited to various weaknesses. Due to their sequential nature, they cannot work in parallel. As a result, they cannot take full advantage of GPUs and TPUs.

The popularity of DL nowadays mainly arises after the availability of numerous annotated datasets and computational resources that facilitate parallel processing through modern Single Instruction-Multiple Data (SIMD) hardware accelerators such as GPUs and TPUs. In addition, GPUs and TPUs make processing faster, cheaper, and more powerful.

Transformers represent the current state-of-the-art Natural Language Processing models for analysing text data.

Based on these premises, the project leverages the most recent pre-trained transfer learning models applicable to email messages to explore their full potential to prevent phishing attacks. Transformers have successfully overcome several limitations thanks to a mechanism called Attention. Additionally, Transformers made it easier to process text data in parallel rather than sequentially (thus improving execution speed). Finally, transformers today can be easily implemented in Python thanks to the *Hugging Face* Library and *TensorFlow Hub*. Further details about the methodology adopted will be provided in Section 5.

The data analytics methodology follows the traditional CRISP-DM framework.

The models proposed will be assessed using conventional metrics like *accuracy*, *Recall*, *precision*, and the F_k score, as well as computing time and GPU resources, which will be tracked and evaluated against other models and methodologies described in the literature review.

[1] https://www.cs.cmu.edu/~enron/

Figure 1 shows a graphical representation of the model pipelines proposed.

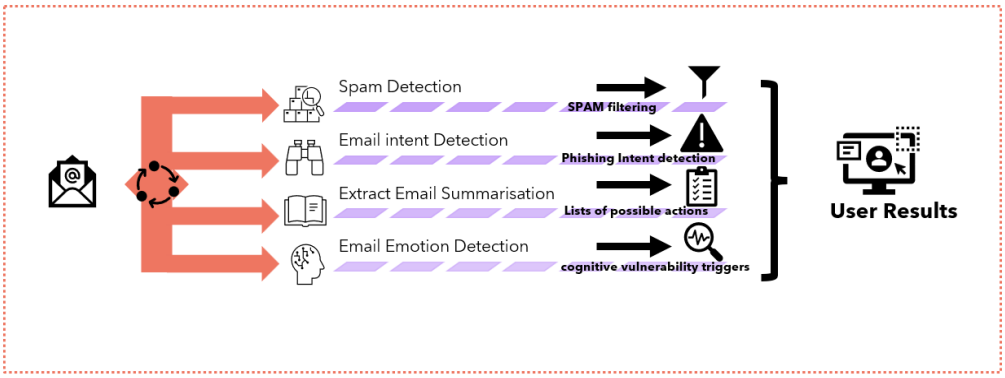

Figure 1: Model Pipeline

4 Design Specification

The solution proposed is located in a wider Cyber Security Awareness program in everyday organisational life. In this view, theoretically, the advantages of our human-centric solution are numerous, including:

1. Open to all: able to reach the entire company population, regardless of the role held within the organisation.
2. Continuous: able to maintain great attention to cyber threats over time.
3. Compatible: able to have a minimal impact on time and professional commitments.

The solution proposed will be a real-time training tool pointing to affect the behaviour of the entire company; this involves the ability to "engage" participants, overcoming historical limits in training in the business environment. Figure 2 illustrates the various project components and data flow.

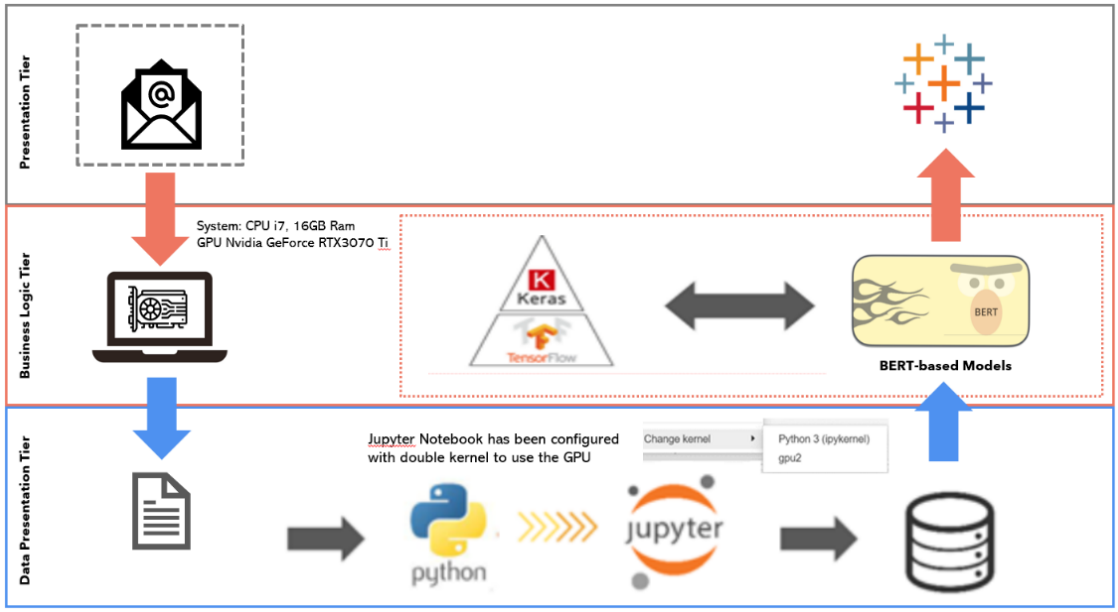

Figure 2: Solution Design Tiers

Since we will run heavy computing tasks like training DL models and fine-tuning BERT-like models, it is necessary to use an environment with GPU or TPU.
We have created a GPU-based environment in *Jupyter Lab* configured to use GPU from *Anaconda*®.
The system also includes a *Bokeh*®[2] server to monitor models' performance, CPU, GPU and memory usage in real-time.
For the scope of the project, Python language is used, along with the libraries *Keras*® and *Tensorflow*™. Keras is a Python-based Deep Learning API developed to facilitate rapid experimentation using TensorFlow Machine Learning. The Keras platform also includes a variety of high-performance pre-trained models.
The Google AI team developed a transformer-based pre-trained model called Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). The BERT base architecture and some small variants are used for this research.
The programming languages, tools, and libraries utilised throughout the project are listed in Table 1.

Table 1: Setup

IDE	*Jupyter Lab*
Computation	GPU
Type	Nvidia 3070Ti
Number of GPU	1
Programming language	Python
Framework	Tensorflow, Pytorch
Modelling library	SimpleTransformer, Tensorflow Hub, Hugging Face, Transformer, Sklearn, Pandas, Numpy, Matplotlib, Seaborn

5 Implementation

5.1 Spam detection

Spam emails are unwanted emails shared in bulk intending to gather data, do phishing, perform social engineering, and start an attack. Usually, these emails follow a pattern, advertisement, and marketing messages, most of which are filtered by the email firewall.
This filtering method is called *Spam Classification* and uses a model trained on the "spam", "not spam" dataset. Spam detection is a text classification problem, a common task in NLP.
We compare our proposed models against current popular machine learning methods like *Xgboost*, *Random Forest* and *SVM*, and *ANN*.
The *Enron*[3] dataset is a set of emails collected from the workplace which suits the project's scope. For this branch, we use the Enron dataset[4] labelled for spam detection.

[2] https://developer.nvidia.com/blog/gpu-dashboards-in-jupyter-lab/
[3] http://www.aueb.gr/users/ion/data/enron-spam/

5.1.1 Data Understanding

The Enron-Spam dataset for spam classification was collected by V. Metsis, I. Androutsopoulos and G. Paliouras and described in their publication *"Spam Filtering with Naive Bayes - Which Naive Bayes?*". The dataset, as illustrated in Figure 3, contains a total of 10.000 spam and 9.441 non-spam ("ham") email messages (19.441 emails total). (Metsis, 2006)

Klimnt and Yong have conducted an extensive description of the dataset. (Yang, 2004)

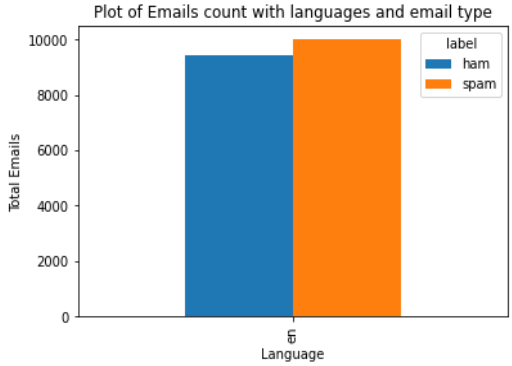

Figure 3: Plot of Email Classes Spam and Ham

5.1.2 Data Preparation

Data Preparation is a crucial part of building a model; data quality determines the model's performance. This phase describes the data, the selection of data with the criteria for inclusion and exclusion, eventual data cleaning, data integration and eventually data augmentation if required.

The email files are organised in folders containing only "ham" or "spam" files. Finally, the data are loaded into the pandas' data frame.

Our training classifiers use transformer-based pre-trained models, so we avoid traditional text pre-processing techniques, such as stopword removal, stemming, and lemmatisation, to preserve semantic meaning. Instead, each sentence of raw data is tokenised before being fed to the pre-trained models using tokenisers specific to the pre-trained model.

5.1.3 Data Pre-processing

The pre-processing phase consists of generating pandas' data frames from the Enron dataset.

We have generated the features in an unsupervised way using the TF-IDF algorithm and then used these features to train Models on labelled Enron data; this approach has given promising results in related works. EDA techniques show that data distribution is quite balanced between spam and ham features, and the token length frequency in both subsets didn't show a significant discrepancy.

Because this is a binary classification problem, we have used the simplest approach to reduce the majority class to balance the dataset. A column language has also been added; while in this case, we only have English text, it can be useful in provision to extend the work in multi-lingual data.

We use 85% for training and 15% for validation, as in previous related work.

We also train a classic 3-layer NN model in *Keras*, a *Random Forest*, *SVM* and *XGboost* model for comparing purposes.

5.1.4 Modelling

Based on the literature review, various machine learning and deep learning models have been investigated. Finally, we decided to adopt for spam detection technique of transformers and transfer learning with the scope to obtain better evaluation at less cost. Furthermore, transformer blocks have been demonstrated to be state-of-the-art in NLP tasks since 2018 using BERT. Our model, deriving from BERT, has a similar training procedure consisting of two stages. BERT derives from a first pre-training and is trained to predict masked words in a sentence with large-scale datasets. First, in the pre-processing stage, BERT removes any noise and duplicate records in the dataset. Then, since any model only works with numeric data, in the encoding process, BERT converts each sentence into embedding vectors, separates the words, and adds special tokens, [*CLS*] and [*SEP*].

The pre-trained model is then fine-tuned in the second stage with a labelled target dataset for a specific classification task.

5.1.5 Model Training

The first DL model created consists of a three-layer model with the assistance of *Keras* and the *TensorFlow* library. Layer 1 consists of 512 neurons with *Relu* activation, layer 2 of 256 neurons with *Relu* activation, using a dropout 0.5 to prevent overfitting. Because this is a binary classification problem, we use binary cross-entropy as the Loss function.

The classic ML and DL models are then used as a baseline to compare them with our proposed solution on traditional evaluation metrics, F_1 score and execution time.

First, we imported *TensorFlow* packages and loaded several Bert models using the *TensorFlow hub* library: classic *BERT*, *DistilBERT*, *Small Bert*, *Universal-sentence-encoder*, *Albert*, *Expert* and *Electra*.

- *Small BERT* is one of the smaller pre-trained BERT variants.
- *Universal-Sentence-encoder* is built upon the base model, employs a 12-layer BERT transformer architecture and can be used for text classification.
- *Albert* is another lite version of BERT.
- *Expert* is a collection of BERT models pre-trained on specific datasets and tasks to improve performance. The utilised model is initialised from the base *Wikipedia + BooksCorpus* BERT model and fine-tuned on the *SQuAD 2.0* dataset.
- *Electra* is pre-trained, resembling a generative adversarial network (GAN). Electra (like BERT) computes dense vector representations for natural language using a deep neural network with the Transformer architecture.

The approach chosen for the classification problem is a *functional* approach contrary to a *sequential* function for a more robust and flexible model.

First, the model is initialised using the *Keras_layer_input* method. Then, *Bert_encoder* is responsible for encoding the pre-processed text into embedding vectors. The neural network is then initialised with two layers, *Drouput* and *Dense*, to prevent overfitting. In this case, we use 0.1% and pass the output of the best layer as a function.

In the *Dense* layer, we use *sigmoid* as the activation function, so the prediction probability is between 0 and 1. During the compiling stage, we test the model for various *optimiser* and *loss*

function values also considered in related work. The models use *Adam* optimiser to improve model performance and reduce error. The loss function is *binary_crossentropy* to calculate the model errors because the output can be 0 or 1. The last stage is fitting the model, so the model learns from the training dataset and gains knowledge. We have tested the models on 16 epochs so that they will iterate 16 times. Since we use *sigmoid*, we have set the threshold to 0.5 to round the result to 0 and 1. The final stage is the evaluation stage, which includes testing the models.

5.2 Text Summarisation

As mentioned earlier, summarisation can help users to identify the most important content of an email body without feeling overwhelmed, providing the essential information to understand the purpose of the email. We use *Enron Email* and *BC3 Corpus* datasets for summarising email text. *BC3 Corpus* contains human summarisations that can be used to calculate the ROUGE metric to evaluate the accuracy of the summarisation task

5.2.1 Data Understanding

The original *Enron Email Dataset*[5] provides *150,000* emails from *150* users sent in a corporate environment, and the *BC3 Email Corpus*[6] contains 40 email threads.

The *BC3 Email Corpus* contains human summaries that can be used to test the accuracy of machine learning-generated summaries. The Enron email dataset lacks human summaries but can be used to demonstrate the practical application of automatic summarisations to a huge number of emails.

The raw *Enron* email dataset is a *Maildir* directory with folders separated by employee names containing the emails. Enron emails were sent using the *Multipurpose Internet Mail Extensions 1.0* (MIME) format. Keeping this in mind helps find the correct libraries and methods to clean emails in a standardised fashion.

Exploring the datasets can go a long way to building more accurate machine learning models and spotting any possible issues with the dataset. In addition, since the Enron dataset is quite large, we can speed up some of our computations using *Dask*[7].

5.2.2 Data Preparation

Data preparation follows several steps, extract the useful feature from each email, convert the email list into pandas' data frames and clean the data.

The dataset is pre-processed iteratively on each email to extract the columns used to train the model. Pre-process phase includes the classic steps like removing whitespaces, removing images, remove stopwords. In addition, pre-trained models require receiving raw data converted into a suitable format using tokenisers.

[5] https://www.cs.cmu.edu/~enron/
[6] https://nlp.cs.ubc.ca/bc3.html
[7] https://docs.dask.org/en/stable/

5.2.3 Data Pre-processing

In the first stage, it cleans both *Enron* and *BC3* datasets to perform email text summarisation. An additional dataset is needed since the Enron corpus lacks summaries, even if the corpus is more comprehensive. The BC3 dataset is split into two XML files. One contains the original emails, and the other includes the summarisations created by the annotators. Each email may contain several summarisations from different annotators.

Two data frames are loaded. One has the wrangled original emails containing several pieces of information, such as thread identifier, email sender, recipient, title, and body email. The second data frame contains human summarised emails. So, first, we create a data frame for both XML files; then, we join them together using the thread number combined with the email number to create a single final data frame.

The pre-processing phase consists of cleaning the data. Since the Enron data was collected in May 2002, according to Wikipedia, it's strange to see emails past that date; reading some of them seems to suggest it's mostly spam. On the other hand, there appear to be emails dated exactly *1980-01-01* that we decided to drop since, without the true date, we won't be able to understand where the email fits in the context of a batch of summaries. Unfortunately, the raw Enron email Corpus tends to have many unneeded characters that can interfere with tokenisation, so we decide to perform some cleaning.

We remove stopwords that don't provide additional sentence meanings like 'and', 'or'; this is applied to both the Enron and BC3 datasets.

After the cleaning stage, emails' body sentences are tokenised, and data stored in *.pkl* files, respectively, *enron_df.pkl* and *BC3_df.pkl*.

5.2.4 Modelling

E*xtractive summarisation* is the methodology adopted to identify and concatenate the most important email sentences. For the project scope, we rely on different variants of BERT to perform extractive summarisation showing their results on the Enron email dataset. As proved in various papers, a flat architecture with inter-sentence Transformer layers performs the best, achieving state-of-the-art in the context of article summarisation, for example, on *CNN/Dailymail* and *NYT* datasets. (Mike Lewis, 2020)

Using Transformers' parallelisation further enhances training speed and uses GPUs and TPUs to maximise training efficiency. Transformers are pre-trained using pre-trained language models (PTLMs) on huge corpora, and their knowledge is transferred to downstream tasks such as ATS. Therefore, PTLMs' semantic and contextual features can enhance the quality of the resulting summaries by utilising their universal and rich embeddings.

We will use different methods for extractive summarisation, including classic *TextRank*, to compare Transfer Learning models, such as *BertSum*, *BART*, *DistilBERT*, and *T5*.

5.2.5 Model Training

For this branch, we compare several methods for extractive summarisation. The first model built is *TextRank*; the second is *Bert-Classifier* to generate summaries of someone's inbox email. TextRank uses sentence embeddings to determine sentence similarity to determine the most important sentences in a corpus. We use word embeddings from both *word2Vec* and the

embeddings determined by *BERT*[8]. Based on the content that both phrases contain, TextRank estimates the degree of similarity between them. This overlap is computed as the number of shared lexical tokens divided by each sentence's length and are examples where the TextRank algorithm is proposed. A second method built uses the *BertSum* model, a fine-tuned version of the BERT model specifically for extractive text summarisation. (Liu, 2019) The algorithm can also generate summaries of someone's inbox over a period. The BC3 Corpus contains human summaries that can be used to generate ROUGE metrics to understand the summarisations' accuracy better.

Our proposed model leverages pre-trained models; in this way, instead of building and fine-tuning an end-to-end NLP model, we can take advantage of the pre-trained model to generate our new word vectors. In particular, we build further models, such as *T5*, *BART* and *DistilBERT*, simply using the pipelines available on *Hugging Face*: this makes it easy to summarise and has very low consumption. For example, *BART* is a model pre-trained in English and fine-tuned on the *CNN Daily Mail* dataset. (Mike Lewis, 2019)

BART is a transformer model (seq2seq) based on a bidirectional encoder (BERT) and autoregressive decoder (GPT). BART is pre-trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text.

The particular checkpoint we use for this branch has been fine-tuned on *CNN Daily Mail*.

Due to computational limitations and the time-consuming task, we only perform summarisation and evaluate it over a subset of 50 emails.

The models built are compared using different percentages of 10%, 25%, and 50% words regarding the length of the generated email summary. The most promising result has set the summary length to 25% of the original email length.

Since the BC3 Corpus contains human summarisation, it can be used to calculate *ROUGE* metrics. ROUGE metric helps to understand how accurate the summarisation is.

The ROUGE metric is an evaluation metric used to test machine-generated summaries against a human "Gold standard". The leading TextRank summary currently has a ROUGE 1 F_1-Score of 31.74%. (Nath, 2022)

5.3 Intent Analysis

With the help of this branch, a recipient can determine the danger of an email by highlighting and contextualising suspicious portions of the email in conjunction with the information in other branches. Additionally, pointing out potential phishing intent could train recipients to identify phishing attempts.

Email is one of the top-consuming activities in the workplace. Since the advent of more sophisticated phishing attacks, industries have invested in research beyond categorising emails by keywords. As well as studying how to prevent and be more effective and how to be more proactive. (Krishnan, 2020) (Sproull, 1991)

[8] https://github.com/UKPLab/sentence-transformers

5.3.1 Data Understanding

Our analysis relies on the email corpus presented in (Prabhakaran, 2012) from the original Enron corpus, which has been manually annotated for various forms of power relations. In total, 122 email threads are contained in the corpus, which includes 360 messages and 20,740-word tokens. Email threads in this collection have been selected from a version of Enron's email corpus, with some missing messages restored (Harnly, 2006).

The corpus we use also contains manual annotations of dialogue acts, as described in (Hu, 2009). We use these annotations to model the communication thread's dialogue structure.

Each message is divided into Dialog Functional Units (DFUs) to segment the thread. Dialogue Act (DA) labels are assigned to each DFU: *Request for Action*, *Request for Information*, *Inform*, *Inform-Offline*, and *Commit*.

A further methodology adopted for the specific task is based on (Ala Mughaid, 2022). On a corpus containing over 55,000 "sent" emails from the Carnegie Mellon[9] archive, the emails were tagged with "click" and "download" keywords.

5.3.2 Data Preparation

To avoid the possibility of counting the same email multiple times when sent to a list of recipients, we only use "sent" emails. In addition, phishing emails are sent emails, which provides a common comparison point. It is common for replies and forwards to have several emails, so they are longer emails. Additionally, replies and forwards may contain multiple headers. Thus, these emails contain more substantial information than phishing emails, increasing the probability that it is not phishing emails.

Because the data is also unbalanced, we have decided to perform data augmentation using *DistilBERT* on specific classes, avoiding down sampling. (Qianhao Yu, 2022)

5.3.3 Data Pre-processing

We used the *Parakweet Email Intent Dataset*[10] to prepare this batch as a baseline. These data come from the same dataset of *Enron* emails and are used to train and test the ability to recognise sentences in emails revealing "intent". The creators of this dataset, Parakeet Lab, follow the same (Hu, 2009) definitions for a request, proposal and commit and defines "intent" as one of these three speech acts. For the intent, we consider mainly the following acts:

- *Request-Action:* requesting an action, task, meeting, information, or favour from another person.
- *Inform:* a directive or order.
- *Commit:* commit to a meeting, task, action, or delivery.

Due to highly unbalanced classes, we decided to consider only the *Request-Action* and *Inform* classes. Therefore, data augmentation is undertaken using the *nlpaug* library (Qianhao Yu, 2022) to balance the two selected classes. To produce the synthesised data, we generate sentences in which words are differently spelt, synonyms are added, words are embedded, and contextually embedded words are created. As a result, artificial email bodies appear as natural

[9] https://www.cs.cmu.edu/~enron/

[10] https://www.cs.cmu.edu/~enron/

as real ones. The main difference between data augmentation and oversampling is that the first add variations to the input, whereas the second cannot alter it. (Varad Pimpalkhute, 2021). Figure 4 shows the balanced classes after applying the data augmentation technique.

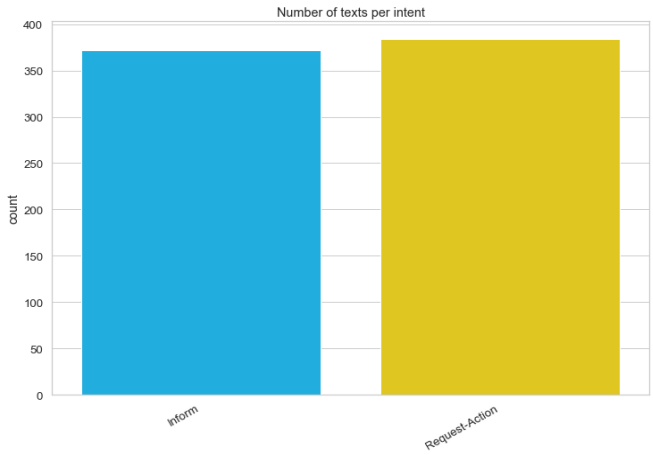

Figure 4 Balanced Classes *Inform* and *Request-Action* as a result of data augmentation.

We also decided to avoid downsampling on the test set because it will artificially bias the metrics for evaluating the model's fit, which is the point of the test set.
Using the *NLTK* library (Bird and Loper, 2004), we filtered the content of the emails organising it into a list of its sentences, creating a dataset with all the processed sentences of all the emails, resulting in a total selected unique 390 sentences, and used these for labelling.
Then, using the *sklearn library* (Pedregosa, 2011), we split the dataset into a training set (80%) and a testing set (20%).
After data augmentation, our dataset includes 624 labelled sentences, 319 sentences labelled as *Inform* and 305 sentences labelled as *Request-Action*.
Additionally, short descriptions (tags) have been added to emails to categorise phishing intents, such as "clicklink" or "downloadfile". Being a multilabel classification problem, we use *sigmoid* as the activation function of the last layer and binary cross-entropy as the loss function. That is because we consider the classification of each label to be independent of all the other labels. Therefore, using *softmax* or Categorical cross-entropy is wrong in this scenario.
We also use *Keras* Categorical accuracy instead of the basic *sklearn.metrics.accuracy_score()*; the accuracy_score function in multilabel classification mode only considers a sample as correctly classified when all the true labels and predicted labels for that sample match each other.

5.3.4 Model Training

To cherry-pick phishing intent in new emails, we train models that recognise malicious intent in their sentences. If other indicators, including cognitive triaging or the summary, are suspicious, users should pay close attention to phishing intent in new emails.
We'll use pre-trained models available on *TensorFlow Hub* for the intent recognition branch. Loading models from TensorFlow Hub, there are multiple BERT models available: *BERT-uncased*, *Small-BERT*, *DistilBERT*, *ALBERT*, *BERT-Experts*, and *Electra*, which have fewer parameters and can be fine-tuned much faster. In addition to the mentioned models, there are

multiple larger versions, which can provide even increased accuracy, but they cannot be fine-tuned on a single GPU.
We load the models from TensorFlow Hub, so we can choose which BERT model to load and fine-tune easily. The BERT-based models return a map with three important keys: *pooled_output*, *sequence_output*, and *encoder_outputs*. We use the *pooled_output* array for fine-tuning, creating a very simple fine-tuned model with the pre-processing model, the selected BERT model, one *Dense* layer and a 0.1% *Dropout* layer. Because this is a binary classification problem, we use *binary cross-entropy* loss and *Adam* optimiser on five epochs.
Besides this binary classification problem, we have also developed a second approach, using short tag descriptors to identify speech-act tags that are usually used in phishing emails, such as "click" or "download".
The models built are then compared with classic models such as *SVM*, *Random Forest*, and *XGBoost* used in the literature for text classification (Annalisa Occhipinti a, 2022) and to classify intent on the Enron dataset. (Cohen, 2004)

5.4 Emotion Recognition

We developed this branch of emotion classification based on a brilliant paper on cognitive triaging. (Allodi, 2019) As highlighted in the literature, phishing emails can be triggered by six cognitive triggers: reciprocity, consistency, social proof, authority, liking, and scarcity. (Ferreira, 2015)
The six principles of persuasion(PPSE) defined by Robert Cialdini are today widely used in the marketing field:

- Reciprocity: people always tend to return a favour, almost feeling obliged to someone.
- Commitment and consistency: if people commit themselves, they will more likely honour their commitment.
- Social proof: people tend to think that the actions that others do are right; in case of doubt, we tend to consider what other people have done.
- Authority: people tend to follow the order of an authority or an authoritative person (or presumed to be such) in a given field.
- Liking: people are easily persuaded by other people they know and like.
- Scarcity: scarcity generates demand; if it is rare, it seems more attractive.

Based on these principles, **some phishing techniques have been developed and analysing them is helpful to discover the information that the hacker aspires to in an ultimate attempt to hack the victim's brain**.

5.4.1 Data Pre-processing

The dataset[11] consists of 343 sentences of a random sample sentence from emails labelled under six classes.
Due to unbalanced classes, data augmentation using the *nlpaug* library (Qianhao Yu, 2022) is undertaken to balance the minority classes. To produce the synthesised data, we generate

[11] https://git.imp.fu-berlin.de/piel82/social-engineering/-/blob/master/dataset_finished.csv

sentences in which words are differently spelt, synonyms are added, words are embedded, and contextually embedded words are created using DistillBERT. As a result, artificial sentences will appear as natural as real ones. Figure 5 shows the classes result after data augmentation.

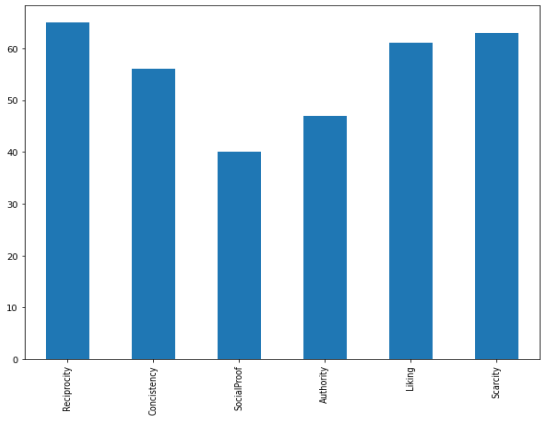

Figure 5: Classes Distribution after Data Augmentation

To preserve the semantic contents of the sentences, we are not employing conventional text pre-processing techniques such as stop word removal, stemming, and lemmatisation because we are using transformer-based models to train the classifier. Instead, the raw data is transformed into an appropriate format before being presented to pre-trained models using tokenisers suited for the pre-trained model.

5.4.2 Model Training

The final dataset is very small, consisting of 403 sentences, but BERT is pre-trained on vast amounts of text. Moreover, it has an unsupervised objective of masked language modelling and next-sentence prediction. It can be fine-tuned with various task-specific goals even on small datasets giving discrete results (Artzi, 2021).

In this scenario, transfer learning is preferable because there is not much-labelled training data. The general idea is to use knowledge learned from tasks for which a lot of labelled data is available in settings where only little labelled data is available. Moreover, creating labelled data is expensive, so optimally leveraging existing datasets is key. We perform a common text classification task using the Keras library for this branch. Finally, we will present our approach and compare our models' results with baselines from previous works.

For the baseline models, we follow two approaches:

- The first approach represented one label with a dense layer with multiple neurons as output.
- Our second approach included creating separate dense layers with one neuron per label.

Results show that, in our case, a single output layer with multiple neurons works better than multiple output layers.

Neural network models can be configured to support multilabel classification and work well, depending on the specifics of the classification task. As Neural networks can directly support multilabel classification simply by specifying the number of target labels present in the problem and the number of nodes in the output layer, we specify six dense layers output and 0.1% dropout to prevent overfitting.

Each node in the output layer uses sigmoid activation; this will predict a class membership probability for the label, so a value between 0 and 1.

6 Evaluation

6.1 Spam Detection

The results from the simple deep learning model do very well on the test data. The results from other models are close. *XGboost* also performs very well. However, neither Random Forest nor SVM have been optimised beyond the defaults. As a result, they may achieve better performance with tuning. BERT- like models give the most enthusiastic results, shown in Table 2. In particular, *Universal Sentence Encoder* has outperformed SOTA on the Enron dataset. (Tida, 2022)

Table 2: Spam Detection Model Results

Model	**Metrics**				
	Accuracy	**Precision**	**Recall**	**F_1 score**	**Time(sec.)**
Random Forest	0.87	0.86	0.86	0.86	10.10
SVM	0.91	0.92	0.90	0.91	45
ANN	0.97	0.97	0.98	0.97	10.5
XGBoost	0.89	0.89	0.89	0.89	11.2
BERT	0.96	0.96	0.96	0.96	46
DistilBERT	0.95	0.95	0.95	0.95	42
Small Bert	0.96	0.96	0.96	0.96	34
Universal Sentence Enc.	**0.99**	0.99	0.99	**0.99**	46
Albert	0.93	0.93	0.93	0.93	38
Expert	0.92	0.92	0.92	0.92	48
Electra	0.95	0.95	0.95	0.95	34

6.1.1 Discussion

The performance of transfer learning is notable since the model's parameters are pre-trained in the first stage with large-scale data and transferred to a smaller dataset, so fine-tuning is required just on the small, labelled dataset. Moreover, the easy usage of the models makes them very appetible for a more complex task.

We have tested the models using both Sigmoid and Relu activation functions to prevent vanishing gradient problems. Both have given equivalent results. Relu has the advantage of being less computationally expensive than Sigmoid and has been used in related works (Tida, 2022)

Traditional metrics are used to evaluate the model proposed, such as accuracy, precision, Recall, and the F_k score, as well as computational time and GPU resources, which have been compared with other models and techniques presented in the literature review.

6.2 Email Summarisation

As previously mentioned, the evaluation metric used is the *ROUGE* score, which is very common for evaluating summarisation performance. Despite this, various factors have to be taken into account. From the literature review and the project experience, we agree that summarisation performance cannot be measured exclusively with metrics. Factors like *fluidity* and *quality* are subjective to the context in which the summary is performed and require a human inspection too. For example, ROUGE doesn't assess how fluent the summary is. Instead, ROUGE only evaluates the adequacy by counting how many n-grams in our generated summary match the n-grams in the reference summary (or summaries, as ROUGE supports multi-reference corpora). While Table 3 shows the ROUGE results for each model, Table 4 illustrates the text summarisation result from the BART model on a phishing email example provided by the repository of Berkeley University.[12]

Table 3: Text Summarisation Model Results

Model	**Rouge-1**		
	F_1	**Recall**	**Precision**
TextRank	0.27	0.20	0.38
TextRank(BERT)	0.31	0.28	0.35
BertSum	0.26	0.41	0.19
BART	**0.37**	0.40	0.35
DistilBERT	0.32	0.39	0.28
T5	0.31	0.39	0.26

Table 4: Email Summarisation Example

Original Email	'Hello! My name is Shafaq. Your website or a website that your company hosts is infringing on a copyright-protected images owned by myself. Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get the evidence of my copyrights. Download it right now and check this out for yourself: hxxps://sites.google.com/view/a0hf49gj29g-i4jb48n5/drive/folders/shared/1/download?ID=308682351554855915 I believe you have willfully infringed my rights under 17 USC Section 101 et seq. and could be liable for statutory damages as high as $150,000 as set forth in Section 504(c)(2) of the Digital Millennium Copyright Act ("DMCA") therein.'
Extractive Summarisation With BART	'Your website or a website that your company hosts is infringing on a copyright-protected images owned by myself. Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get evidence of my copyrights. Download it right now and check this out for yourself: "link."

6.2.1 Discussion

Because ROUGE is based only on the content overlap, it can determine if the same general concepts are discussed between an automatic summary and a reference summary.

[12] https://security.berkeley.edu/education-awareness/phishing/phishing-examples-archive

Still, it cannot determine if the result is coherent or if the sentences flow together sensibly. High-order n-gram ROUGE measures try to judge fluency to some degree. From a Rouge-1 performance perspective, BART has been chosen for this purpose since it has the most encouraging results.

With T5, discrete results have been obtained without the need for fine-tuning for individual tasks. Therefore, this model has made implementing the purposes relatively easy.

Although achieving the above might seem useful, there are no objective criteria for determining how content should be summed up in an ideal scenario. Therefore, we evaluated the summarisation at different limits, such as 10, 20, and 50 per cent.

However, closer examination and metrics such as cross-validation are potentially needed to determine best what words limit the summary should have.

A quarter of the size of the original email has given the most promising result. However, it was still based on subjective criteria such as grammar and readability.

Even if some numerical results are not exciting from a metric perspective, we only accept them from a mathematical point of view as we know the limit of statistical evaluation on a complex task such as text summarisation. Therefore, we decided to compare and analyse the model performance on some real phishing emails to evaluate the models' capability in the specific context, such as producing a summary containing the psychological triggers and intent from the other tasks.

6.3 Intent Detection

Table 5 shows the overall results from the models built for intent detection on speech act tasks.

Table 5: Intent Detection Model Results

Model	**F_1**	**Accuracy**	**Recall**	**Precision**
Random forest	0.82	0.80	0.77	0.83
SVM	0.81	0.79	0.78	0.82
XGBoost	0.81	0.78	0.77	0.82
LGBMClassifier	0.80	0.78	0.78	0.80
Small BERT	0.77	0.95	0.74	0.81
DistilBert	0.85	0.97	0.81	0.82
Universal Sentence Enc.	0.84	0.85	0.82	0.58
Albert	0.86	0.95	0.97	0.80
Expert	**0.86**	**0.96**	**0.82**	**0.86**
Electra	0.80	0.96	0.81	0.78

Expert-BERT shows the best results in terms of F_1 score and *accuracy*.

Testing the model Expert on the usual email used for the summarisation task, we obtain the result shown in Table 6.

Table 6: Intent Detection with *Expert*

Intent Detection with *Expert*	**Intent**
'Hello! My name is Shafaq. Your website or a website that your company hosts is infringing on a copyright-protected images owned by myself.'	Inform
Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get the evidence of my copyrighs	Request-Action
Download it right now and check this out for yourself:	Request-Action
I believe you have willfully infringed my rights under 17 USC Section 101 et seq. and could be liable for statutory damages as high as $150,000 as set forth in Section 504(c)(2) of the Digital Millennium Copyright Act ("DMCA") therein.	Inform

Table 7 presents the results from the models built for intent detection on the speech-act tags, "click-link" and "download". Finally, Table 8 provides the model results and its ability to detect the speech-act tags in the email body.

Table 7: Speech Acts Model Results

Model	**F_1**	**Accuracy**	**Recall**	**Precision**
Albert	**0.95**	**0.94**	0.96	0.95
DistilBERT	0.95	0.93	0.95	0.94
Expert	0.65	0.32	0.80	0.56
Electra	0.85	0.64	0.98	0.75
Small-BERT	0.94	0.92	0.97	0.94
Talking-Heads	0.94	0.90	0.98	0.91

Table 8: Speech Acts Example

Speech Acts tags Recognition example	**Tags**
'Hello! My name is Shafaq. Your website or a website that your company hosts is infringing on a copyright-protected images owned by myself.'	Neutral
Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get the evidence of my copyrighs	Click-Link
Download it right now and check this out for yourself:	Click-Link
I believe you have willfully infringed my rights under 17 USC Section 101 et seq. and could be liable for statutory damages as high as $150,000 as set forth in Section 504(c)(2) of the Digital Millennium Copyright Act ("DMCA") therein.	Neutral

6.3.1 Discussion

Among traditional models, the best result is given by Random Forest in the Enron intent classification task. There are two main reasons why Random Forest over Gradient Boosted Decision is working better, and they are both related:

- RF is much easier to tune than GBM
- RF is harder to overfit than GBM

However, it is generally true that a well-tuned GBM can outperform RF. Also, as Tianqi Chen highlights, RF has traditionally been easier than parallelism. However, that is no good reason anymore, given there are efficient ways to do it with GBMs. (Chen, 2019)
The results have shown that the most promising models for implementing this branch have been *Expert* and *ALBERT*, edging out *DistillBERT* slightly for *accuracy* and *loss*.

6.4 Emotion Recognition

As shown in Table 9, Small-BERT has achieved the best results for this branch regarding F_1 score and *Recall*. Table 10 provides the model results and its ability to detect the emotion triggered by the email sender.

Table 9: Emotion Recognition Models Results

Model	**F_1**	**Accuracy**	**Recall**	**Precision**
Albert	0.31	0.36	0.40	0.28
DistilBERT	0.44	0.61	0.25	0.16
Universal	0.28	0.26	0.28	0.16
Electra	0.33	0.33	0.18	0.30
Small-BERT	**0.53**	**0.57**	**0.51**	**0.58**
BERT-uncased	0.23	0.38	0.18	0.52
Expert	0.16	0.35	0.11	0.46

Table 10: Emotion Recognition Example

Emotion Recognition Example	**Emotion**
'Hello! My name is Shafaq. Your website or a website that your company hosts is infringing on a copyright-protected images owned by myself.'	Consistency
Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get the evidence of my copyrighs	Social Proof
Download it right now and check this out for yourself:	Scarcity
I believe you have willfully infringed my rights under 17 USC Section 101 et seq. and could be liable for statutory damages as high as $150,000 as set forth in Section 504(c)(2) of the Digital Millennium Copyright Act ("DMCA") therein.	Liking

6.4.1 Discussion

As the dataset is small, it is good practice to repeatedly evaluate the neural network models on the same dataset and report the average performance between repetitions due to the learning algorithm's stochastic nature.
As mentioned, better results have been achieved relying on the Small-BERT pre-trained model. To further improve the model performance, we can use k-fold cross-validation to get an unbiased estimate of model performance when making predictions on new data. K-folder cross-validation has several benefits, including using all the available data, more metrics in order of K to evaluate model performance and achieving better precision.

With this in mind, in future, we will evaluate using repeat K-fold cross-validation with ten times and three repeats; this means that the model will predict three probabilities for each sample. These can be converted to clear class labels by rounding the values to 0 or 1. We can then calculate the accuracy of the classification for clear class labels.
For the project scope, the aim was to raise awareness about the possible phishing indicators.
A combined probability score could also be generated from the results that indicate how likely a phishing email is based on one or more of the six cognitive triaging qualities. For example, from previous studies, *authority* is the most used persuasion principle in phishing attacks, as shown in Figure 6, so we can assign those emails a higher score. (Keith S. Jones, 2021)

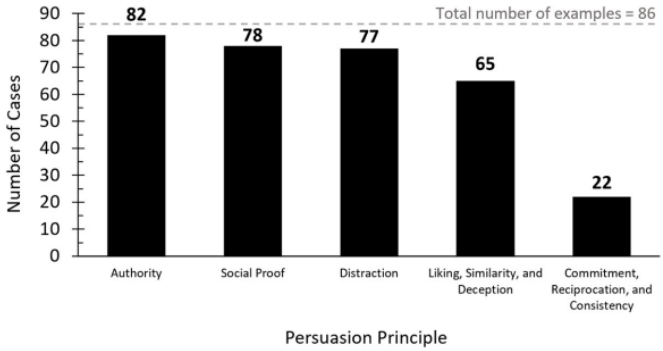

Figure 6: Frequency usage of Persuasion Principles in Social engineering (Keith S. Jones, 2021)

As a result of the combined assessment from the other branches, users can make rational and informed decisions based on all the information available. For example, informing users of *Request-Action* intent and generating tags such as "Click-link" and "Download" with the prospective danger of being too many has the aim to alert the user of a possible phishing attack, despite whether the email has been recognised as spam or not. Table 11 shows the result of the four branches of the phishing email used for the summarisation branch to provide an example result from the four pipelines.

Table 11: System-Generated Result

Spam Detection	**SPAM**		
Extractive Summary:	'Your website or a website that your company hosts is infringing on a copyright-protected images owned by myself. Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get evidence of my copyrights. Download it right now and check this out for yourself: "link."		
Phishing Indicators	**Intent**	**Tags**	**Emotion**
Take a look at this document with the links to my images you used at website.berkeley.edu and my earlier publications to get the evidence of my copyrighs	Request-Action	"Click-Link"	Social proof
Download it right now and check this out for yourself:	Request-Action	"Click-Link"	Scarcity

7 Conclusion and Future Work

The research project aimed to answer the main question of how machine learning could help employees detect phishing emails. Moreover, we decided to base our research on the most recent Transformer and Transfer learning techniques to promote the context-aware approach highly required today.

Using email to extract psychological triggers, malicious intent, and concise summaries of emails has helped us develop a human-centric notification mechanism. Finally, we present the information above to the user in a meaningful way to better understand the continuously evolving phishing patterns.

An examination of the effectiveness of this methodology on a larger scale will be possible through user studies and objective experimental analysis. To determine whether an email is a phishing attack, we can also use a concerted metric of the trigger and intent density.

From a learning perspective, the research project has allowed us to exploit the many types of pre-trained models available and benefit from two aspects:

- There are noticeable performance improvements for all aspects of computation: graph/eager mode and CPU/GPU devices monitored through the *Bokeh* server.
- We benefit from *TensorFlow Hub*, a repository of trained machine learning models ready for fine-tuning and deployable anywhere.

Moreover, the proposed solution's modularity is flexible and scalable, enabling businesses to adopt multiple techniques simultaneously.

Optimising the obtained result further will require larger-scale datasets and experiments, as the available data has been the main limitation. In future work, we will aim to improve the results obtained using a larger corpus like *Avocado Research Email Collection*, which has proven to give optimal results in similar problems. Furthermore, the model designed enables us to enrich the solution with further branches, for example, Named Entity Recognition (NER), Person of Interest (POI) and even tailor the tool to specific user attributes, like age and gender.

In conclusion, we want to highlight that the solution proposed doesn't replace existing phishing filters. Instead, the ultimate purpose is not to simply detect phishing emails but to educate employees and present the information that allows them to self-learn advanced malicious patterns.

References

Adhika Pramita Widyassari, S. R. G. F. S. E. N. A. S. A. A. D. R. I. M. S., 2022. Review of automatic text summarisation techniques & methods. *Journal of King Saud University - Computer and Information Sciences*, 04 Apr, 4(34), pp. 1029-1046.

Afonso Mendes, S. N. S. M. Z. M. A. F. T. M. S. B. C., 2019. Jointly Extracting and Compressing Documents with Summary State Representations. *Cornell University.*

Ala Mughaid, S. A. A. H. S. T. A. A. &. E. A. E., 2022. An intelligent cyber security phishing detection system using deep learning techniques. *Cluster Comput.*

Alkhalil, Z., 2021. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Front. Comput. Sci.*, 09 Mar.

Allodi, A. v. d. H. a. L., 2019. Cognitive Triaging of Phishing Attacks. *28th USENIX Security Symposium (USENIX Security 19)*, pp. 1309--1326.

Annalisa Occhipinti a, b. L. R. a. C. A. a., 2022. A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications,,* Volume 201.

Anon., 2020. A Neural Named Entity Recognition and Multi-Type Normalisation Tool for Biomedical Text Mining. *IEEE,* pp. 73729-73740.

Artzi, T. Z. F. W. A. K. K. Q. W. Y., 2021. REVISITING FEW-SAMPLE BERT FINE-TUNING.

Ashish Vaswani, N. S. N. P. J. U. L. J. A. N. G. L. K. I. P., 2017. https://arxiv.org/abs/1706.03762?context=cs.

Bekkerman, R., 2004. Automatic Categorisation of Email into Folders: Benchmark Experiments on Enron and SRI Corpora.

Bhargava, A., Celikyilmaz, A., Hakkani-Tür, D. & Sarikaya, R., 2013. Easy contextual intent prediction and slot detection.

Carmona-Cejudo, J. M. et al., 2011. GNUsmail: Open framework for online email classification.

Cohen, W. W., 2004. Learning to Classify Email into {"}Speech Acts{"}". *Association for Computational Linguistics,* pp. 309-316.

Danqing Wang, P. L. M. Z. J. F. X. Q. X. H., 2019. Exploring Domain Shift in Extractive Text Summarisation.

Dey, R. & Salem, F. M., 2017. Gate-variants of Gated Recurrent Unit (GRU) neural networks.

Drucker, H., Wu, D. & Vapnik, V., 1999. Support vector machines for spam categorisation. *IEEE Transactions on Neural Networks,* 10(5).

Ferreira, A. C. L. L. G., 2015. Principles of Persuasion in Social Engineering and Their Use in Phishing. *Human Aspects of Information Security, Privacy, and Trust.*

Frank Cremer, B. S. M. F. A. N. K. M. M. F. M. &. S. M., n.d. Cyber risk and cybersecurity: a systematic review of data availability. *The Geneva Papers on Risk and Insurance - Issues and Practice,* Volume 47, pp. 698-736.

Harnly, J.-y. Y. a. A., 2006. Email thread reassembly using similarity matching. *In In Proc. of CEAS.*

Hashemi, H. B., 2016. Query Intent Detection using Convolutional Neural.

Hazarika, A. B. ·. S. M., 2016. Machine Learning for Email Spam Filtering: Review,. *Cryptography and Security.*

Hu, J. a. P. R. J. a. R. O., 2009. Contrasting the Interaction Structure of an Email and a Telephone Corpus: A Machine Learning Approach to Annotation of Dialogue Function Units. *Association for Computational Linguistics,* p. 357–366.

Huntley, S., 2020. *Findings on COVID-19 and online security threats.* [Online] Available at: https://blog.google/threat-analysis-group/findings-covid-19-and-online-security-threats/
[Accessed Apr 2022].

Hussain, M. M. H. M. H. M. W. H. W., 2017. Header Based Spam Filtering Using Machine Learning Approach. *International Journal of Emerging Technologies in Engineering Research,* pp. 133-140.

Irwin, L., 2022. *The 5 most common types of phishing attack.* [Online] Available at: https://www.itgovernance.eu/blog/en/the-5-most-common-types-of-phishing-attack

Ivano Lauriola, A. L. F. A., 2022. An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. Jan.

Jacob Devlin, M.-W. C. K. L. K. T., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Jiang, Z. & Wang, Z., n.d. MTL-DAS: Automatic Text Summarisation for Domain Adaptation. , Volume 2022.

Jingli Shi, L. H. G. W. W. L. &. Q. B., 2022. Focus-Based Text Summarisation with Hybrid Embeddings. *Lecture Notes in Computer Science book series (LNAI,volume 13151)*, Mar, pp. 705-715.

K.Mohameda, W. S.-K. R. A., 2021. Automatic text summarisation: A comprehensive survey. *Expert Systems with Applications*, Mar, Volume 165.

Keith S. Jones, M. E. A. a. M. K. T., 2021. How social engineers use persuasion principles. *Information and Computer Security*, 23(1).

Kraut, L. A. D. E. K. E., 2006. Email overload at work: An analysis of factors associated with email strain.

Krishnan, S., 2020. Exploitation of Human Trust, Curiosity and Ignorance by Malware. *researchgate.*

Lampert, A., 2010. Detecting Emails Containing Requests for Action.

Lila Carden, J. V. K. M. F., 2021. Enhancing human resource management in process improvement projects. *Organizational Dynamics*, 50(2).

Liu, Y., 2019. Fine-tune BERT for Extractive Summarization. *Institute for Language, Cognition and Computation.*

Lukas Halgas, I. A. J. R. C. N., 2019. Catching the Phish: Detecting Phishing Attacks using Recurrent Neural Networks (RNNs). *Cryptography and Security*.

Mantrach, F. K. M. A. M. A. G. a. 5. a. M., 2015. Evolution of Conversations in the Age of Email Overload. *Conference: the 24th International Conference*.

Matt Balogh, W. B. D. P. a. M. A. K., 2022. Understanding the management of personal records at home: a virtual guided tour. *Information Research*, Volume 27.

Matthew E. Peters, S. R. N. A. S., 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks.

Matwin, S. K. a. S., 2000. Email Classification with Co-Training.

Metsis, V., 2006. Spam Filtering with Naive Bayes – Which Naive Bayes?.

Mike Lewis, Y. L. N. G. M. G. A. M. O. L. V. S. L. Z., 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

Miller, D., 2019. Leveraging BERT for Extractive Text Summarization on Lectures. *Computation and Language*, Jun.

Nallapati, R. Z. B. d. S. C. G. Ç. X. B., 2016. Abstractive text summarisation using sequence-to-sequence RNNs and beyond. *The SIGNLL Conference on Computational Natural Language Learning*.

Narasimhan, A. R. a. K., 2018. Improving Language Understanding by Generative Pre-Training.

Nikita Benkovich, R. D. D. G., 2020. DeepQuarantine for Suspicious Mail. *CEUR Workshop Proceedings 2479*.

Patil, P. D., 2014. An Overall Survey of Extractive Based Automatic Text Summarisation Methods. *International Journal of Science and Research* .

Pedregosa, F., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*.

Prabhakaran, V. a. N. H., 2012. Annotations for Power Relations on Email Threads. *Proceedings of the Eighth International Conference on Language Resources and Evaluation ({LREC}'12)*, pp. 806-811.

Qianhao Yu, X. Z., 2022. Application of Data Augmentation in Financial Sentiment Analysis Task. *SSRN*.

Radev, D. R. a. H. E. a. M. K., 2002. Introduction to the Special Issue on Summarisation. *Computational Linguistics*, Volume 28, p. 399–408.

Research, S., 2022. *Most common delivery methods and cybersecurity vulnerabilities causing ransomware infections according to MSPs worldwide as of 2020.* [Online] Available at: https://www.statista.com/statistics/700965/leading-cause-of-ransomware-infection/
[Accessed 27 Jul 2022].

Rong Xu, M. K. S. M. Y. H. M. A. D. M. P. a. A. G. M. P., 2006. Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts. *AMIA Annu Symp Proc*, pp. 824-828.

Sewani, S. M. A., 2005. Analysing Behaviorial Features for Email Classification.

Sproull, L. &. K. S., 1991. Connections: New Ways of Working in the Networked Organization.. *Administrative Science Quarterly*.

Stamp, T. S. F. D. T. K. P. M., 2022. Convolutional Neural Networks for Image Spam. *Information Security Journal: A Global Perspective*, Apr, pp. 103-117.
Technology, I. G., 2014. *IBM Security Services 2014 Cyber Security Intelligence Index,* s.l.: IBM Global Technology Services.

Theodore Longtchi, *. R. M. R. *. L. A.-S. A. A., 2022. INTERNET-BASED SOCIAL ENGINEERING ATTACKS, DEFENSES. Aug.

Tida, V. S., 2022. Universal Spam Detection using Transfer Learning of BERT Model. *Hawaii International Conference on System Sciences.*

Varad Pimpalkhute, P. N. a. T. D., 2021. *Proceedings of the Sixth Social Media Mining for Health ({\#}SMM4H) Workshop and Shared Task*. Mexico City, Mexico: Association for Computational Linguistics.

Vasima Khan1, *. T. A. M., 2021. Pretrained Natural Language Processing Model for Intent Recognition (BERT-IR). *Human-Centric Intelligent Systems*, 1(3-4).

Victor Sanh, L. D. J. C. T. W., 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Vidal, Z. A. a. J., 2018. To Read or To Do? That's The Task.
Wagner, G. H. a. A. C. a. L. G. a. M. S. a. V. P. a. S. S. a. G. M. V. a. D., 2019. Detecting and Characterising Lateral Phishing at Scale. *28th USENIX Security Symposium (USENIX Security 19).*

Xuanyi Dong, 2018. *Cornell University.*

Yang, B. K. a. Y., 2004. The Enron Corpus: A New Dataset for Email Classification Research. *springer*.

Zainab Alkhalil, w. H. w. N. a. w. K., 2021. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Front. Comput. Sci.*, 09 Mar.

Zhang, K., 2022. Focus on the Action: Learning to Highlight and Summarise Jointly. *Association for Computational Linguistics*, pp. 4095 - 4106.

Zhenzhong Lan, M. C. S. G. K. G. P. S. R. S., 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.