A Comparison of Machine Learning Algorithms for Zero-Shot Cross-Lingual Phishing Detection

by

Dakota Staples

Bachelor of Computer Science, UNB, 2022

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisors:	Saqib Hakak, PhD, Computer Science
	Paul Cook, PhD, Computer Science
Examining Board:	Rongxing Lu, PhD, Computer Science, Chair
	Haruna Isah, PhD, Computer Science
	Zhen Lei, PhD, Civil Engineering

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

August, 2023

Abstract

Phishing is a major problem worldwide. Existing studies have focused mainly on detecting emails in one language (mostly English). However, detecting emails in multiple languages is challenging due to a lack of datasets. Without ample data from which to learn, the models cannot detect a benign email from a spam email accurately, resulting in false positives and negatives. This research aims to compare the performance of numerous machine learning models and transformers using zero-shot learning for multilingual phishing detection. In a zero-shot learning set-up, the model is trained on one language and tested on another. English, French, and Russian emails are used as the training and testing languages. My results show that, on average, XLM-Roberta performs the best out of all the tested models in terms of accuracy scoring 99% testing on English, 99% testing on French, and 95% testing on Russian.

Dedication

To my family.

Acknowledgements

First and foremost, I am thankful to my supervisors, Dr. Saqib Hakak and Dr. Paul Cook, for their guidance, insight, and encouragement throughout this research. Their expertise and mentorship have been instrumental in shaping the direction of this thesis. Not only have they helped with this research, but taught me how to perform and write about research at a high standard.

I would also like to thank all members of the Faculty of Computer Science who have taught me and helped me along the way whether directly or indirectly.

And I devote an enormous amount of thanks to all of my friends and all of my family who have supported me through my classes and research.

Lastly I would like to extend thanks to the University of New Brunswick and the Faculty of Computer Science for providing funding to my research and making it possible.

Table of Contents

List of Tables

List of Figures

List of Symbols, Nomenclature or Abbreviations

GPT	Generative Pre-Trained Transformer
SVM	Support Vector Machine
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term Frequency-Inverse Document Frequency
FFNN	Feed-Forward Neural Networks
TREC	Text Retrieval Conference
EMFET	E-mail Features Extraction Tool
SMOG	Simple Measure Of Gobbledygook
XLMR	XLM-RoBERTa
ISO	International Organization for Standardization
En	English
Fr	French
Ru	Russian
Zh	Chinese
Vi	Vietnamese
De	German
Hi	Hindi
Ur	Urdu
Fa	Persian
Es	Spanish
Lt	Lithuanian
P	Precision
R	Recall
RCNN	Region-based Convolutional Neural Networks
NLP	Natural Language Processing
FFNN	Feed Forward Neural Networks
LSTM	Long Short Term Memory
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
IWSPA-AP	International Workshop on Security and Privacy Analytics

Chapter 1

Introduction

The prevalence of phishing emails in today's society is a significant concern resulting in a multitude of negative consequences for those who fall victim. Phishing emails can grant hackers access to various accounts such as email, social media, and e-commerce sites. These malicious emails can also lead to the download and execution of malware, which can cause significant damage. As email is widely used, whether for personal or business purposes, phishing is a severe issue affecting a large number of people [12].

Phishing is a global problem and not limited to English-speaking countries. In the year 2021, the French public service site issued a warning against phishing emails [8], while the Spanish police tweeted an alert about a phishing scam in the year 2022 [79]. It is reasonable to assume that phishing scams exist in various languages. Therefore, phishers would be more likely to send emails in a language with which the recipient is more familiar, increasing the likelihood of success.

As phishing does not appear in only one language, it is important for a phishing detection system to be able to detect in many languages. Furthermore, English is also not the only language in the world and many other languages are also spoken and written. In fact, it is estimated there are approximately 6000 languages in the

world [72]. An attacker in Russia would be highly unlikely to send phishing emails to other Russians in English. This is because Russian is the majority language spoken in Russia, therefore making it more likely that the victim will understand and fall for the scam. Although there is no academic literature existing to support this, the existence of the Russian and French phishing emails used in my experiments [57] can be conjectured to support this claim. If French scammers only sent phishing emails in English, then how could these French emails find their way into an end-user's mailbox and subsequently be reported? This claim intuitively makes sense, and is supported by the existence of this data.

Phishing attacks are increasing at an alarming rate, causing considerable damage to both businesses and consumers. According to CyberTalk, an estimated 6 billion phishing attacks were predicted to occur in the year 2022 alone, with users opening 30% of phishing emails [61]. In the year 2021, phishing was the second most expensive type of attack to users and businesses, costing them an average of 4.65 million USD. IBM found in the same year that 1% of all emails are phishing emails. In addition, 1% of phishing emails reach the user's inbox if using Microsoft 365 Outlook, meaning they are not filtered by a spam or junk filter. The Federal Bureau of Investigation (FBI) IC3 (Internet Crime Complaint Center) has even issued a public service announcement to businesses in the past, warning them about phishing emails, which have caused a loss of 43 billion USD between June 2016 and December 2021 [17]. Phishing emails are the most significant type of attack, with reports increasing from 25,244 in the year 2017 to 323,972 in the year 2021 [26]. These statistics are not limited to the United States of America (USA), as 177 countries have reported business email compromise attacks [26].

1.1 Problem Statement

Given the prevalence of phishing and the scale of the problem worldwide, it is essential to have countermeasures to prevent and mitigate it. This research examines numerous machine learning models and transformer language models (such as GPT-3) which are tested in various settings to examine how models perform at multilingual phishing detection. To the best of my knowledge, this is the first work of its kind where GPT-3 has been tested for multilingual phishing email detection. To perform tests, data in English, French, and Russian was acquired [57] and will be discussed further in the thesis. English, French and Russian were chosen to be the languages as this is the data that was most readily available. To test the models, I examine how each model performs in a cross-lingual zero-shot setting, as well as a monolingual setting for comparison. A cross-lingual zero-shot setting refers to training on one language and testing on another that it did not see during training.

1.2 Research Objectives

This research aims to answer the following research questions.

- **R1:** Do transformer language models outperform traditional machine learning models at monolingual detection?
- **R2:** Do transformer language models outperform traditional machine learning models at cross-lingual zero-shot detection?
- **R3:** Does a large language model (GPT-3) perform better than an existing multilingual transformer language model (XLM-RoBERTa) in this task?
- **R4:** Does training on multiple languages improve detection capabilities over a single language?

To answer these questions, two main experiments are performed. The first experiment is a monolingual setting where the models are trained and tested in the same language. The second experiment is a cross-lingual zero-shot setting where the model is trained in a language (or languages) that is different from the language of its testing data in a zero-shot setting. The first experiment aims to answer my first research question, while the second experiment aims to answer the remaining questions two through four. For all experiments, English, French, and Russian are the languages that will be used to train and test the models in the various experiments.

It is found that while transformer models do generally outperform traditional models at monolingual detection in terms of accuracy, albeit only by a small margin, there are other factors that should be considered in this case such as environmental costs and training time. Furthermore, when examining cross-lingual zero-shot settings, transformers perform better at this task by a large amount in terms of accuracy. The results also show that GPT-3 is not better than XLM-RoBERTA at the zero-shot multilingual phishing detection task. Finally, it is found that training on multiple languages improves detection when compared to training on a single language.

1.3 Organisation of Thesis

The organisation of the thesis is as follows. Chapter 2 discusses the related and previous works and contains a discussion on phishing emails. Chapter 3 contains the experimental setup. The results are discussed in Chapter 4. Chapter 5 concludes the work.

Chapter 2

Background

This chapter aims to provide a background for the thesis which includes a definition of phishing, discussing the tactics used in phishing, the types of phishing relevant to this research, the structure of an email, and existing studies on monolingual and multilingual phishing email detection models.

2.1 Phishing Definition

The term phishing is believed to have originated around the year 1995 [45] and has since become extremely prevalent in the world. The word phishing is purported to have to come from fishing and phreaking, a form of hacking against telephone systems[42].

One consistent definition of phishing is impossible to obtain and many researchers have different ideas about what exactly characterizes phishing. Lastdrager found in their research that there was at least 113 definitions of phishing showing it is not an easy term to define[45]. However, there are common themes which were identified which led the author to describe phishing as a means of deception in which impersonation is used to obtain information and can be scaled. To expand upon this, not only is the goal to obtain information, but also to maliciously affect the

victim in any way as the attacker could craft a phishing email containing malware as an attachment. We can see differences in definitions not only in research, but in government organizations as well. The Canadian Centre for Cyber Security defines phishing as using phone calls, texts, emails, or social media to trick the user[27]. The FTC (Federal Trade Commission) in the USA however, only lists emails when it comes to phishing as a definition, but also warns users against pop-ups[19]. Furthermore, NIST (National Institute of Standards and Technology) defines phishing as being performed through websites or emails[55]. These American organizations, FTC and NIST, give similar definitions.

Although it is important to understand that phishing does not have one concrete definition, it is also important to define phishing from the viewpoint of this research. For the purposes of this research, phishing will only refer to email phishing and will be defined as "phishing is an email based means of deceiving users for an attacker's malicious purposes." This is because my work only examines multilingual phishing email detection. My scope does not include websites, text messages, pages, or messages on social media, phone calls, or any other medium which has previously been defined by researchers, government, or websites. However, my definition is still broad enough to cover the various sorts of emails which may be researched. Whether the attacker is looking to deceive an end user out of money, to obtain their information, to infect them with malware, or any other purpose, my definition is still accurate.

2.1.1 Phishing Tactics

Phishing emails themselves have evolved and changed to get around detection mechanisms and to better fool their targets, but their underlying tactics remain the same. The below-mentioned six tactics are not just for phishing emails but remain true for most social engineering attacks. Social engineering is a general term that refers to attacks against a user. These typically involve social interaction but that is not

a necessity. This could take the form of dumpster diving for sensitive information. However, most social engineering attacks work by social interaction[20].

2.1.1.1 Intimidation

This tactic is simply scaring the target into giving the attacker the desired data. Intimidation often contains direct threats, forcing the victim into giving the attacker what they need out of fear [20]. This could be in the form of an account "lockout" which needs their password to be removed; otherwise, their data will be lost. Another form this could take is threatening to release private information about the victim unless the attacker gets what they want [38].

2.1.1.2 Consensus

Consensus is making the victim think that since everyone else is doing what the hacker asks, they should, too [20]. A typical attack here may be tricking the user into downloading a malicious application. On a website, a hacker may write fake positive reviews themselves using multiple accounts or with the use of bots. This way, when the victim sees the application, they are more compelled to download it, as other users seemingly have used it and liked it. In a phishing email, the attacker may include the malicious application as a link or direct the victim to a website to steal their data or have them download the malware there.

2.1.1.3 Authority

An attack using authority is when the attack involves using a position of power over a victim to get results [20]. Impersonating a boss of a company to get user passwords is an example of an attack using authority. This is related to intimidation, as the target will feel scared to react negatively in fear of consequences. So, typically an attack using authority will use intimidation as well, and it is an effective combination

for hackers to use.

2.1.1.4 Scarcity

Scarcity is frequently used in marketing and not just in social engineering attacks [49]. When marketing agencies say "only 10 products left at this sale price", people are more likely to purchase them. This can also be leveraged in attacks [20]. An attacker could craft a phishing email telling the victim that they were selected to be part of a special test group for a program and only five spots are left in it. So, the user is compelled to download it without considering the potential consequences and, subsequently, potentially infect themselves. This often ties into urgency as well, as the victim is encouraged to act quickly before the deal or spots are gone.

2.1.1.5 Urgency

Urgency can be used in attacks by making the user act quickly so that they do not think about whether the scenario makes sense or not [20]. An example would be a hacker saying that the account is locked and the password is needed to unlock it within an hour, otherwise, the account will be permanently deleted. Social engineering attacks that involve urgency always want the victim to act very quickly so that they do not have time to think or contact others about it.

2.1.1.6 Familiarity

Familiarity is when an attacker relates the situation to something that the victim is familiar with and comfortable with [20]. The attacker may impersonate a relative or close friend, or they may do research through social media to make it seem like they are familiar with you. These scams can ask for money for hospital bills or bail fees. This being said, there are many variations that use this tactic.

Phishing may use some of the above tactics or use all of them in a very complex

phishing email to deceive users. Moreover, some tactics align better together than others like intimidation, urgency and authority, or scarcity and urgency. In addition to tactics, phishing also needs targets, which can influence what kind of phishing attack takes place.

2.1.2 Phishing Email Attacks

Phishing attacks are often general with the hackers sending emails out to anyone they can in as wide a range as possible to increase the likelihood that an email will be successful [59]. However, under my definition there exists two subcategories of phishing which are more targeted. These are:

1. *Spear Phishing*: Spear phishing tailors the attack to a group or organisation [59]. A spear phishing attack may target a single department at a business or target numerous low-level employees at a company. The defining factor in spear phishing is that the phishing emails are personalised to the victim or victims who are not prominent members of the organization.

2. *Whaling*: Whaling is when the target is a high-level executive or public target at a company [59]. These include media managers, C-level executives, presidents, and more. These emails are often highly personalised and contain a lot of information about the targets and their businesses. This type of attack is potentially the most dangerous due to the positions the targets hold. These people often have high levels of access within an organization or the authority to authorize large sums of money transfers and therefore, whaling can have disastrous effects.

2.2 Email Structure

In this section, I will highlight the features of an email that attackers exploit to execute different attacks.

Table 2.1: Summary of an email with the features and description of the feature.

Feature	Description
Header	Information about the sender, routing details, date And more
Body	Main text of the email
Attachment	File attached to the email that may not always be present
URL	Link inside the body of the email which may not always be present
HTML\JavaScript	Code and markup inside in the body of the email which may not always be present

Emails have a distinct composition that allows detection systems to target a specific structure. Some of these structures are optional; however, some are not. When detecting phishing, it is the content that must be analysed and not the structure or absence of a structure itself. This is because structurally, spam and ham will be identical as they are both emails, just differing in content [34]. The structure and various features of the structure are shown in Table 2.1.

The headers of an email, like the email structure itself, contain both mandatory and optional fields. Every header must have the date, the sender's address, and the recipient's address. Additionally, it will contain the routing path the email took to arrive at the recipient's inbox, meaning which domains and IP addresses it went through. One common field in the headers which is optional is the subject line. Although optional, it typically appears. Headers may also contain content type, authentication news, and more [22, 89].

The body of the email contains the main content. This is where the message to the recipient is found. This can be empty and the message contained all in the subject line, but this is uncommon. The body is also where most detection systems aim to analyse. Although headers can be useful, especially for manual analysis, for automated detection such as machine learning, the body contains the main information needed to make the decision. A lot of detection systems analyse the URL provided in an email; seeing if the link and what the link displays match, whether the link is an IP address, or whether it contains hexadecimal. Not every phishing email contains a

URL, however, and therefore it would be naive to think that a system based only on URL detection would work. Some features from the body include the count of "." characters, how many domains are in the email, and if the email contains HTML or JavaScript code [92, 90].

Emails can also contain attachments that can potentially be used; however, because the systems used to detect phishing train and detect based on email text, a file will not be much use. However, a rule-based system may choose to use an attachment, and it certainly aides in manual reviews. For example, a manual reviewer could scan the file with an antivirus system such as VirusTotal. A rule-based system may choose to allow all docx (Microsoft Word files) and zip files while banning Python and Java files [13].

2.3 Monolingual Phishing Email Detection

Monolingual phishing detection is when the system is tasked with detecting only one language. This could be English, it could be Chinese, or another language as well. This would mean the model was only trained on one language as well. If the model is trained on English, French and Chinese, and subsequently tested on English, this would be a multilingual setup. So, monolingual phishing detection must train and test on only one language. In this section, I identify three types of phishing detection: rule based, traditional machine learning, and deep learning. I also examine monolingual phishing detection in languages other than English as the majority of research done in this area is performed in English. A summary of monolingual works is provided in Table 2.2.

Table 2.2: Summary of related monolingual models on phishing email detection where each row contains what metrics the experiment used as well as the dataset(s) and model(s).

Reference	Acc	P	R	F1	Dataset	Model Used
[36] 2020	✓	✗	✗	✓	Enron	BERT
[25] 2019	✓	✓	✓	✓	Enron, IWSPA-AP, and More	THEMIS
[77] 2020	✓	✓	✓	✓	Nazario and csmining dataset	Random forest
[92] 2016	✓	✓	✓	✓	Nazario	Random forest
[74] 2022	✓	✓	✗	✓	PhishTank, Enron and More	Rule Based
[2] 2019	✓	✗	✗	✗	PhishTank, Yahoo, Common Crawl and More	Rule Based
[90] 2022	✓	✓	✓	✓	Ethio Telecom	SVM
[76] 2021	✓	✓	✓	✓	Custom Dataset	LSTM
[83] 2020	✓	✓	✓	✓	Enron, IWSPA-AP and More	THEMIS and BERT
[82] 2020	✓	✓	✓	✓	Enron, IWSPA-AP and More	THEMIS
[80] 2021	✓	✗	✗	✗	Enron and a Private Dataset	bi-LSTM

2.3.1 Rule Based Methods

Starting with a basic approach which is not as current or state-of-the-art as deep learning or machine learning, one method of phishing detection is a rule-based approach and thus should still be considered and reviewed. SatheeshKumar et al. detailed one such approach, called SniffPhish 2.0 [74]. In this work, they have listed 20 rules and features that phishing emails typically follow. One rule was simply whether the URL was an IP address, while another rule was how many dots and subdomains are in the URL. In their experiment, they obtained websites from PhishTank, Enron Email Dataset [37], and IsitPhishing for malicious websites. Alexa Top Sites, Stuffgate Free Online Website Analyser, Yahoo, Google, and Bing were used to obtain legitimate websites. Once obtained, their system was tested, and it was found that their work outperformed SniffPhish by a large margin. Their work had an accuracy of 91.7% and an F1 score of 91.78%, while SniffPhish had an accuracy of 73.95% and an F1 score of 73.51%. For future work, the system must have more features specific to mobile attacks, and machine learning needs to be implemented in order to reduce the false positive rate. It should be noted, however, that while SniffPhish was designed to specifically be a browser extension and will only examine URLs, not all phishing emails will contain a URL. If a phishing email does not contain a

link, the system will fail. With this being said, this survey is focused on email-based approaches only. However, rule-based approaches for phishing that do detect based off of the email alone do not appear to exist at the time of writing.

Another rule based system was designed by Adewole et al. which is another hybrid system [2]. Their system first uses two datasets, dubbed PhishingDataset1 (1353 emails) and PhishingDataset2 (11055 emails) respectively. PhishingDataset1 had 702 phishing emails, 548 legitimate emails, 103 suspicious emails and 10 features available. PhishingDataset2 had 4898 phishing emails, 6157 legitimate emails, and 30 features available. Some of the features include the URL length, whether the URL is an IP address, and whether the website is indexed by Google Index. The authors use JRip and PART to obtain the rules for their hybrid system. From this system, they were able to obtain an accuracy of 94% on PhishingDataset1 and 99% on PhishingDataset2.

It should be noted that there exist other rule based systems to detect phishing emails [6, 51]; however, since my research does not use rule based methods, they will not be reviewed in depth.

2.3.2 Machine Learning Based Methods

A step up in complexity from rule-based approaches is simple machine learning approaches. These include many different algorithms which include, but are certainly not limited to: naive Bayes, support vector machine (SVM), random forest, and logistic regression. Machine learning approaches also need features to train and classify. These features can come from various parts of the email, such as the presence of the word *urgent* or the inclusion of an IP address. Feature selection is an important part of phishing detection, and the optimal features make a large difference in the system's ability to detect a phishing email.

Research by Sonowal looks at generating features, finding optimal features using

the Pearson correlation algorithm, verifying these best features, and justifying the process [77]. These features include word-based features from the subject and body, link-based features, and readability features using eight readability algorithms. To find these features, preprocessing is used, such as converting to lower case and eliminating stop words. It was then found that common keywords for phishing emails include *account*, *update*, *security*, and *important* . For URL features, phishing emails have mismatches in the visible text link and the actual URL. Other URL features include the use of IP addresses, the length of the URL, the count of "." in a URL, and the use of img tags. The algorithms used for readability include automated readability index, Coleman Liau index, Flesch-Kincaid readability test, Gunning Fog index, SMOG index, LIX readability score, and RIX. To select the features, they compared sequential forward feature selection (SFFS) and binary search feature selection (BSFS). BSFS was found to have the best features, providing an accuracy of 97.41% while SFFS had an accuracy of 95.63% and no feature selection had an accuracy of 95.56%.

Features alone will not detect emails, and a system must be built. One piece of research examines the naive Bayes model compared to the SVM technique for phishing detection. This research uses emails from Ethio Telecom, an Ethiopian based telecommunication company [90]. The emails are first preprocessed, which includes the removal of white space, stop words, and punctuation. In addition, the emails are tokenised and stemming and lemmatisation are performed. After preprocessing, the features are extracted, such as URLs containing IP addresses or hexadecimal, differences in the href and the text shown, the number of dots in the domain name, and the presence of JavaScript and HTML. After the features are extracted, the emails go through SVM or naive Bayes and are classified as spam or ham. It was found that SVM performed better than naive Bayes by a small amount, 2% or less per metric. SVM achieved 98.76% accuracy, while naive Bayes achieved 97.51% accuracy. In

every metric, SVM outperforms naive Bayes.

Yasin proposes another model that uses knowledge discovery [92]. This paper describes the process of knowledge discovery and the steps involved in it. In Yasin's model, they preprocess the data by parsing, tokenising, and stemming the words; removing stop words; performing semantic processing using synonymy and hyponymy; and assigning weight to the phishing terms. The dataset used was the Nazario dataset [53]. Features the authors have chosen for their model are counts of ".", usage of hexadecimal and IP address URLs, the use of images as URLs, and the count of domains in the URLs in the email. The article proceeds to describe the classifiers they will test. The research tested the J48 algorithm, naive Bayes, SVM, Multi-layer perceptron, and random forest. It was found that random forest was the best classifier for their proposed method, with a precision, recall, and F1 score of 99.1%. Random forest performed best when the number of trees was set to 30, yielding an accuracy of 98.8%. It is noted that, for future work, this could be enhanced by creating a mechanism that better reflects the term frequency of new emails that are analysed.

2.3.3 Deep Learning Based Methods

Another survey by Salloum looks at detecting phishing emails using natural language processing (NLP) techniques such as a bag-of-words model, syntax features, semantic features, and word embeddings [73]. First, the article defines and examines what phishing emails are and then looks at some common detection features of them. Some of these features include body and subject line characteristics such as phrases, use of HTML, and more. Other features are sender-based features, script-based features, and URL features. The author then finds that many machine learning models have been made for phishing detection, all using different classifiers such as logistic regression, SVM, random forest, and naive Bayes. It was found that

Bernoulli naive Bayes with a term frequency-inverse document frequency (TF-IDF) matrix performed the best, with an accuracy of 96.5% in 0.157 seconds. THEMIS is then examined, which is a deep learning model based on region based convolutional neural networks (RCNN) [25]. Deep learning is a newer advancement over machine learning and does not require such intense feature engineering, which reduces the amount of overhead required for it. THEMIS performed very well with an accuracy of 99.848%; however, it does not work if the email does not have a header. The research finds that while there has been discussion of NLP techniques, there is a focus on machine learning strategies for detection. They found that there is a need to research the use of NLP techniques such as semantic analysis for detecting phishing emails. Finally, more research into the deep learning aspect is also needed, such as recurrent neural networks (RNN) and convolutional neural networks (CNN).

As shown, there are many different models that can detect phishing; however, there is another advancement. This advancement is transformers. This research has applied a transformer-based deep learning model, specifically the BERT (Bidirectional Encoder Representations from Transformers) model, to detect spam phishing emails alongside an older neural network approach [36]. This trains on the Enron data set with an 80/20 train test split. In this model, they preprocess the data by removing stop words, converting lines to lowercase, and removing words with special characters or numbers. They use a bag-of-words model and compare the models with two features, a count vectorizer and TF-IDF. TF-IDF was found to be a better option, so this was used when testing and training. For the deep learning models, the authors use BERT and feedforward neural networks (FFNN). The pretrained fine-tuned BERT model was found to perform slightly better. It was also found that doubling or halving the number of neurons lowered the F1 scores [36].

2.3.3.1 Federated Learning Based Methods

Not much research exists in federated learning in the field of phishing email detection. While other spam federated learning exists, it is not in the form of emails and therefore out of scope. Federated learning is a decentralised form of machine learning in which multiple decentralised edge nodes are trained on local data samples rather than centralised data samples compared to traditional machine learning approaches [91]. Federated learning provides benefits over traditional machine learning models. The main and most prominent benefit of federated learning is its improvements in privacy [18, 52]. While it is not perfect, it improves privacy over traditional machine learning since the data is never shared directly with a central third party server. However, there is not much research in this field and it is still a developing area.

Research by Thapa et al. examines six research questions [83]. The questions relate to balanced data distribution, scalability, communication overhead, client-level perspectives, and asymmetric data distribution. The data used in this research comes from three sources, Enron spam dataset [37], Nazario dataset [53], and IWSPA-AP phishing emails. The paper then looks at choosing a model. The authors examine and compare two models, THEMIS and BERT. The data is preprocessed by extracting headers, cleaning with Beautiful Soup [66], which includes the removal of stop words using the NLTK package, and finally tokenising the data for each model. It was found that federated learning is feasible compared to centralised learning but could not reach the performance of centralised learning. The research then shows that if the number of clients increases, THEMIS has a degradation in performance and the speed of convergences decreases, while BERT has an increase. The researchers demonstrated that the communication overhead per client depends on the model size and that a client can leverage federated learning to improve its performance. Finally, federated learning was found to be highly resilient against asymmetric data sizes, and if the data is highly diverse, the result is model dependent.

More research by Thapa et al. looks at five more questions related to federated learning [82]: if federated learning is comparable to deep learning trained on a central email repository in terms of model accuracy, how the number of clients affects convergence and accuracy, what is the communication overhead, if clients with varying local data sizes can still be learned from, and if a pretrained model can perform better. The data for this experiment is largely the same as the previous work from Thapa, using the Enron dataset, the Nazario dataset, and the IWSPA-AP phishing emails [83]. THEMIS was selected as the model for this experiment. The preprocessing also largely remained the same as the previous experiments, having a step to extract the headers, cleaning using Beautiful Soup, stop word removal, and tokenization. The results of this experiment showed that federated learning is in fact comparable to deep learning in terms of accuracy. Furthermore, increasing the number of clients decreased the time to converge and the accuracy, as shown in [83] with the THEMIS model suffering from the increase in clients. It was also found that the communication overhead is 0.179 GB per client and that we can still learn from clients that vary in local data size. Finally, research shows that transfer learning can be used for a performance boost. It is noted in the end that for a future direction, homomorphic encryption (a form of encryption in which the cyphertext can be computed on without having to first decrypt it) and differential privacy (a privacy standard in which the presence of an individual record or not should not affect the computation on a dataset - typically achieved by adding noise) may be used alongside federated learning. However, homomorphic encryption increases overhead and differential privacy decreases performance.

Sun et al. propose a model called Federated Phish Bowl (FPB) which is comprised of federated learning and long short-term memory (LSTM) [80]. First, the data goes through a processing stage, which consists of eight steps. Step one is to remove any character that is not a letter, then, for step two, every remaining character

is converted to lowercase. Step three is to tokenise the data, and step four is the removal of stop words. In step five, the tokens are lemmatised. The next step is to remove every word with less than two characters and then convert the tokens back to continuous strings. Finally, the strings are mapped to feature vectors. This research uses a global word embedding strategy [60], which is trained on six billion tokens from Wikipedia 2014 and 26 GB of newswire text. For FPB, LSTM is used, which is based on RNNs. FPB uses a five-layer model comprised of three bidirectional LSTM layers, a fully connected layer with 200 neurons, and an output layer with one neuron. For the dataset, the researchers chose to collect emails from Microsoft 365 and used the Enron dataset to balance it out. There were 1188 emails in total with a 90/10 train validate split. They show that FPB outperforms a client learning on its own and that increasing the number of clients degrades the performance. They close by stating that this research was based on synchronous learning and that asynchronous learning may be a good area of research which could improve FPB.

2.3.4 Monolingual Research in Languages other than English

While the majority of research is in English, there are a few pieces of research in languages other than English in the field of phishing detection. There are some that only detect phishing websites [93], which is not the focus of this research and, therefore, is out of my scope. One example of a monolingual detection system in a language other than English was proposed by Siddique et al. [76]. This research aims to detect phishing emails in Urdu. To do this, the researchers took English emails from Kaggle and then translated them into Urdu, with manual corrections being made by the authors. Furthermore, they also changed them to Urdu script instead of Roman script. This dataset was then published online on GitHub for future research use. In this work, they preprocessed the data by tokenising the data,

Table 2.3: Summary of related models on multilingual phishing email detection where each row contains what metrics the experiment used as well as the languages tested and model used.

Reference	Acc	P	R	F1	Languages Tested
[54] 2017	✓	✗	✗	✗	En, Zh, Vi
[5] 2014	✓	✓	✓	✓	En, De, Hi, Ur, Fa
[47] 2020	✓	✗	✗	✗	Not Mentioned
[15] 2015	✓	✓	✓	✓	Es, Fr, En
[65] 2021	✓	✓	✓	✓	Ru, Lt, En

followed by the removal of stop words. After this, the authors performed stemming on the tokens and finally picked and extracted their features. Multiple algorithms, both deep learning and machine learning, were tested. The tested machine learning algorithms were naive Bayes and SVM while the tested deep learning algorithms were LSTM and CNN. It was found that LSTM performed the best with an accuracy of 98.4%.

2.4 Multilingual Phishing Email Detection

While phishing detection has been studied by many researchers and a considerable amount of meaningful research has been done in this field, not as much work has been done in the area of multilingual phishing email detection. In multilingual phishing detection, the system is tasked with detecting emails in a language that differs from the one (or ones) used during training. For example, training on French and testing on English is multilingual. Another example is training on English and French, and testing on Russian.

This section will examine existing work in spam detection which can detect or attempts to detect phishing emails in more than one language. Table 2.3 shows the papers overviewed in this section at a glance.

2.4.1 Rule Based Approaches

Research by Long et al. looks at examining phishing emails in Vietnamese, Chinese, and English. They achieve this by using a rule-based approach based upon Spam Assassin. These rules were then extended and expanded to make them more suitable for a multilingual approach. This was done by extending previous rule-based approaches, as well as using existing Spam Assassin rules made for other languages. While many tests were performed, the best result for the multilingual dataset was when they tested with 100 rules at a threshold of 0.5. At these values, they achieved a spam detection rate of 49.6% and a false alarm rate of 2.9%. The spam detection rate is defined as the number of spam emails detected as spam divided by the total number of spam emails, and the false alarm rate is defined as the number of ham emails detected as spam divided by the total number of ham emails. For future work, it is mentioned that a bigger dataset is needed, as the dataset they used had only 286 emails [54].

2.4.2 Machine Learning Based Models

Researchers from Lithuania have also looked at multilingual phishing email detection. Their research used English, Russian, and Lithuanian. The two non-English languages come from a private dataset and they also use the Spam Assassin corpus [78] and the Nazario Spam Corpus [53]. Before the data was fed into the model, it was cleaned of special symbols, HTML, CSS, and JavaScript, leaving only the raw text. They tested many different models including naive Bayes, random forest, and SVM. It was found that SVM performed the best, achieving an accuracy of 84%. It is noted that for future work, deep learning could be leveraged to give better results [65].

Another piece of research examines multilingual phishing email detection through translation. The languages they attempt to classify are German, Persian, Urdu,

Hindi, and English. To do this, they first obtained emails from the TREC 2007 dataset. They subsequently built their model from a Bayesian filter and performed three different tests. Since the TREC dataset is in English, the authors translated the emails using Google and Microsoft translator. Test one involved not translating the data, so the filter was not trained for non-English data. Test two translated the data at the training stage, so it was trained on non-English data. Finally, test three translated the data at the classification stage so any non-English emails were translated to English before classification. It was found that test three performed best overall for all languages except one. The accuracy scores for each language were: 89.45% for Urdu, 88.60% for Hindi, 90.05% for Persian, 86% for German, and 91% for English. Test one was found to perform extremely poorly for languages other than English and test two performed similarly to test three, being the same or within 2% [5].

2.4.3 Deep Learning Based Models

Researchers at Sophos modified a BERT model into context-aware tiny BERT (CatBERT). Specifically, DistilBERT is used as a base for CatBERT, as it has been pretrained with multilingual text. They first look at feature extraction, which is fed into a model head. These features are whether the message is internal communication (sender and receiver both belong to the same domain), whether the message is an external reply (if the sender and reply to are different domains), and the number of recipients and carbon copies. For data, they use private internal data that consists of 407 161 malicious emails and 3 842 772 benign emails which were all obtained from customer traffic. The maximum token length was set at 128 and this was also the amount of mini batches used. For the optimiser, Adam was used. Furthermore, five epochs were used. CatBERT was found to have a 99.49% $\pm$ 0.0042 true positive rate with a false positive rate of 0.1. This is a good model, as CatBERT downsizes and

is faster than DistilBERT, which is already smaller and faster than BERT [47].

2.4.4 Custom Models

Finally, Bouarara et al. created a new model that aims to detect spam in English, French, and Spanish. This approach is unique compared to other models as it utilises both rule-based and machine learning approaches. The researchers created a new dataset called MSpam, which consists of 1392 ham emails and 685 spam emails. The majority of the dataset is English. This model is called the artificial heart-lungs system (AHLS), which is inspired by biology. In order to do this, they created an algorithm which first checks to see if the email is blocklisted (a list in which known malicious email senders are kept). If the email is not on a blocklist, it proceeds to clean the email and translate it with Google Translate (if it is not in English). After translation, the email is encoded and vectors for the email are obtained. The algorithm then uses a "heart" filter (which utilises naive Bayes to calculate the probability that a new email is spam or ham) and a "lungs" filter (which uses the result from the heart filter and has two filters representing the right and left lung), which are used to determine whether the email is spam or ham. If the email is spam, the sender is added to the blocklist for future analysis. It was found that for all languages, AHLS achieved an accuracy of 89.80% and for the English group only, 94.08% accuracy was achieved. In addition, they showed that their model beats the baseline of naive Bayes, K-Means, and Decision Tree C4.5 [15].

2.5 Summary

As shown, many works in the field of phishing detection exist, both monolingual and multilingual. However, some gaps exist in the research. One direction research could examine is newer large language models such as GPT-3. Another direction would

be to train on two languages at once and observe if this can help detect emails in a third language. These are the open directions that this thesis will aim to address.

Chapter 3

Methodology

In this chapter, we detail the methodology for my research in detecting phishing emails both in a monolingual and multilingual set up. In section 3.1, we discuss the data I used and where it was obtained. Section 3.2 details which machine learning and deep learning models were used in my experiments. Finally, section 3.3 gives details on my experiments such as what was tested and how the experiments were conducted. The entire methodology of my experiments is shown as a flow chart in Figure 3.1.

3.1 Data Used

In this section, I first analyze various existing datasets and the challenges associated with them. In addition, I examine one of the most popular email datasets for phishing detection in depth, the Enron dataset[37]. Following this, I describe the data that I use, and how it is generated.

3.1.1 Multilingual Dataset Analysis

Multiple datasets were used for this research in order to test the various models (logistic regression, random forest, SVM, XLM Roberta, GPT-2 and GPT3). In total,

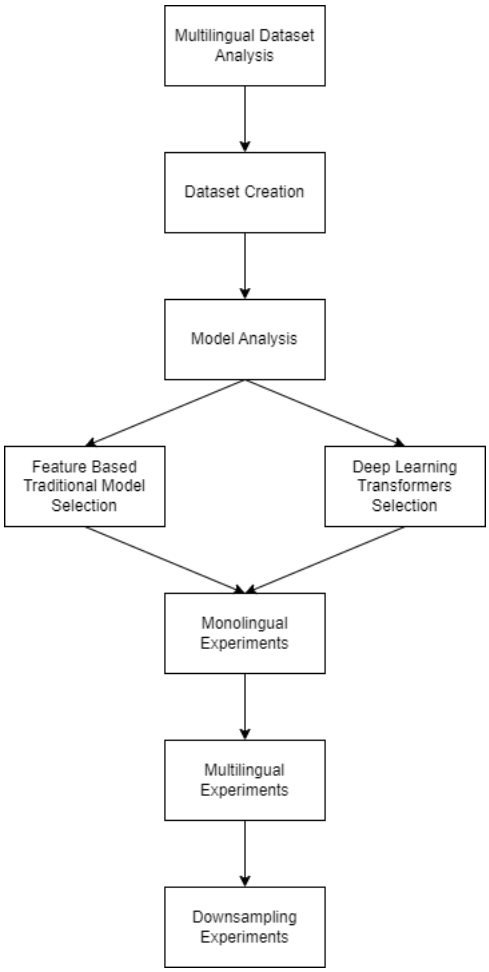

Figure 3.1: Methodology

I trained and tested on three languages which were English, French, and Russian. These languages were selected due to what we were able to obtain within the scope of my research including the budget and time constraints. There is a limited amount of email datasets for phishing detection. The most well-known email dataset, the Enron email dataset, was released in 2004 [37]. The benign emails in this dataset are focused on the Enron Corporation, as the majority are comprised of internal emails being sent. These emails were obtained as part of the investigation and legal proceedings into the Enron Corporation by the Federal Energy Regulatory Commission [43]. It is completely possible that the nature of phishing emails from the Enron dataset and the nature of phishing emails at the time of writing are different due to

not only scams evolving, but language as well. However, this is a largely unstudied area with little research. The one piece of research found suggested the opposite of the claim presented here, that phishing scams have not evolved or that they have gotten better over time; however, this research is from 2015 meaning that phishing could have evolved since then [30]. This was only preliminary research and is an area of study worth investigating further.

There is a negligible amount of phishing datasets in languages other than English. There is a dataset in Urdu, but it is translated data [76]. There is also a dataset containing 200 emails in Spanish, English, and Portuguese [69]. However, this dataset does not have a research paper associated with it and it is difficult to verify the legitimacy of the data. Based on my studies and consulting research articles, two datasets (French and Russian) were obtained from previous work [57] and it was decided that they would be the additional languages based on what could reasonably be obtained for my research. One major advancement in the field would be the creation of a single new multilingual phishing/spam email corpus that is comprised of both spam emails and ham emails, as one currently is unavailable to the public.

3.1.2 Dataset Generation

Table 3.1: Dataset statistics for each language and language pair

Language	Training Data	Testing Data	Combined Ham %	Combined Spam %
English	3983	996	71%	29%
French	472	119	52%	48%
Russian	175	44	50%	50%
English and French	5570	0	89%	11%
English and Russian	5198	0	70%	30%
French and Russian	810	0	50%	50%

For the English data, I used the Enron Spam dataset [50] which is a subset of the Enron dataset [37]. Specifically, the preprocessed version of Enron1 was used. This data was split into a training set and a testing set. The training set contained 3983 emails while the testing set contained 996 emails with 71% of the emails being "ham"

or benign and the remaining 29% being spam.

For the French data, the spam portion of the whole dataset was obtained from previous work done by Pan et al. [57] in which they collected phishing emails from online forums for reporting scams internationally. However, this data only included spam emails and needed to be balanced with legitimate ones. To do this, emails from the TREC07 dataset [1] were translated into French from English using the Helsinki NLP OPUS-MT-train en-fr translation model[84]. This translation model was chosen as it is from a research team from the University of Helsinki, and is highly rated with many downloads. In addition, this is also a specialized model designed only for translation into French from English instead of a general model which can also translate such as T5[63]. Examples of translation are shown below in Table 3.2. Genuine French emails of course would be ideal but data was needed, and if no real French email dataset in a large enough volume exists, then translation must be used. In total, the training set consisted of 472 emails and the testing set consisted of 119 emails with 52% being ham and 48% being spam.

Table 3.2: Email Translation

English Email	**French Translated Email**
I want to produce some boxplots and plot the logged values but have the axis scale in the original, not-logged scale	Je veux produire des boîtes à boîte et tracer les valeurs enregistrées mais j'ai le Échelle de l'axe à l'échelle originale et non guetée
I have a problem with my HP Compaq nx6110 laptop.	J'ai un problème avec mon ordinateur portable HP Compaq NX6110
I am trying to install the gnomeGUI package	J'essaye d'installer le package Gnomegui
I think I should not need to do that	Je pense que nous ne devrions pas avoir besoin de le faire
I've always run jigdo-lite against my own mirror	J'ai toujours couru Jigdo-Lite contre mon propre miroir

For the Russian data, the process worked very similarly to the French data. Spam emails were obtained from Pan et al. [57]. Ham emails were obtained from the Enron Spam preprocessed dataset [50] and were translated into Russian from English using

the Helsinki NLP OPUS-MT-train en-ru translation model [84]. This translation model was also chosen for the same reason as the French translation model, that it comes from a reputable source, is highly rated with many downloads and is a specialized model.

This was the smallest dataset, with 175 emails being used for training and 44 emails being used for testing. The Russian test set was perfectly balanced with 50% for spam and ham.

Finally, I also constructed three more datasets that consisted of two languages that were used for training. The constructed datasets used both the training and testing data from both languages to form one large dataset. I was able to use the testing data here, as I will be testing in another language, meaning that the system will not see any of its training data during testing time. English and French have 5570 emails, English and Russian have 5198 emails, and French and Russian have 810 emails. Table 3.1 lists each language and language pair used, along with its amount of training emails and testing emails, and the combined percentage of ham emails and spam emails.

3.2 Models Used

In total, I tested six models. These models were a part of two different categories, the first being traditional feature-based non-deep learning models and the second being deep learning models which specifically are transformers, which are a type of deep learning model. The code for all experiments including the data used is available at `https://github.com/dks11/MastersThesis`.

3.2.1 Feature Based Models

These consisted of logistic regression, SVM, and random forest. These were chosen to provide breadth across traditional fields, as they all work differently from each other.

To run these traditional methods, the features had to be chosen first. I chose to use E-mail Features Extraction Tool (EMFET), an open source tool to extract features from emails [32]. This tool was tested in research and performed well for their data [31]. I hypothesise that this will work well for me as I am testing in a monolingual set up as explained below, and that features of emails could persist across languages for a multilingual setting. I chose to focus solely on the body of the email in my feature extraction, ignoring the header as not all of my emails have a header attached to them in the data, making my experiment different in this way from the previous research. Unlike the aforementioned research which uses attachments for feature extraction, there are no attachments which EMFET can utilise in my dataset.

There were 59 features focused on aspects of the email body itself, such as the total number of tabs, the total number of periods, and the total count of spam words. Spam words and function words were defined in a list that comes packaged with EMFET when downloaded from the GitHub repository; this list was translated into French and Russian for the purpose of extracting features in those languages. There are also 23 features focused on the readability of the body including the count of simple and complex words and the simple measure of gobbledygook (SMOG) index feature which is an estimate of how many years of education a person needs to understand a piece of text. Finally, there were seven features focused on lexical diversity including hapax legomena, which are words that only appear once in a given piece of text, in this case, an email. While I use 89 features in total (see Appendix A), EMFET provides the ability to examine the header and attachments as well. Using all features, the authors of EMFET were able to obtain 99.3% accuracy

on the Spam Assassin dataset using random forest.

3.2.2 Deep Learning Transformers

Transformers are a type of deep learning model in which the model utilises attention to understand the relationships between tokens, even if they are not close in a given sequence. These models also have an encoding layer that generates hidden representations from an input sequence and a decoding layer that uses the hidden representations to generate the output sequence [86].

The transformers which I tested are XLM-RoBERTa[21], specifically the base version from HuggingFace (an openly available website which provides many free deep learning models), GPT-2 from HuggingFace[62], and GPT-3[16] using the OpenAI API. There is a large version of XLM-RoBERTa available on HuggingFace as well.

As privacy is a concern for users when using a product, I use GPT-2 as it can be downloaded so that a user never has to send data to an outside system. It represents currently what can be done in terms of the official GPT series that everyone can freely access. However, Huggingface has available some open source multilingual large language models such as Bloom [10] which everyone can also freely access. Some models such as Llama 2 by Meta have only become available since this research was completed and the results were obtained [85]. It should be noted that these models are not the only multilingual transformer models available. These experiments detailed below could be run using these large language models (which are transformers like GPT) in future work to observe whether the results have any degradation or improvement compared to the API version of GPT-3 used in this experiment.

GPT-3 represents the best of what can be done most accessibly in the GPT series at the time of the research. However, this is not free on the OpenAI website. In addition, the source code is not available, as is the case with HuggingFace models. I used the API version, specially the "ada" which is the smallest and cheapest model

available. It has been trained on data up to October 2019 and can handle a maximum of 2048 tokens at once. For fine-tuning ada, it costs $0.0004 per 1 thousand tokens when training and $0.0016 per 1 thousand tokens when using the fine-tuned model. GPT-4 is still inviting users to get access to their API and is not open to all. While GPT-4 is not downloadable, it stands to reason that a similar model may exist someday as an openly available Huggingface model.

Finally, XLM-RoBERTa is used to test a model outside of the GPT series and is an improvement over the traditional mBERT model. Introduced in 2019 by Guillaume Lample and Alexis Conneau, XLM-RoBERTa is an improvement over mBERT by having a larger training size, an improved training method, better multilingual representations, and better performance [48, 21]. It is important for my work to consider not only one class of transformer, hence the inclusion of XLM-RoBERTa. In addition, I wanted to include a model specifically meant for multilingual problems that was purposely trained to capture representations between languages. For these reasons, XLM-RoBERTa was chosen. For breadth and future work, other deep learning models such as RNNs [33, 71, 70] and CNNs [46] could be considered for the problem of multilingual zero-shot phishing email detection. However, since these models often require a large volume of training data and not many instances exist especially for French and Russian, and since they are not designed for multilingual settings like XLM-RoBERTa, I chose not to test them in this work.

3.3 Experimental Setup

This section overviews the experiments that I perform to answer the research questions. First, I overview the experiments which do not downsample the data, which I refer to as full data experiments. Then, the downsampling experiments in which a partition of the data is purposefully withheld in order to perform the experiment

are detailed. System specifications for running traditional machine learning models and deep learning models are also discussed.

3.3.1 Full Data Experiments

To conduct my experiment, I use the training and testing sets to perform 12 experiments in which I train on one or two languages and test on another language which may be the language it saw during training in the case of the monolingual experiments. Shown in Table 3.3, these experiments were put into each of my machine learning and deep learning models for training and classification. Experiments using feature-based models were run on the Windows 10 operating system using an i7 processor, 16GB of RAM, and no dedicated GPU, only integrated graphics. Deep learning models were run on the cloud computing company Paperspace using multiple GPUs such as the NVIDIA Quadro P5000, the NVIDIA Quadro RTX 5000, and the RTX A4000. These used Intel Xeon E5-2623 v4 and Intel Xeon Silver 4215R as CPUs. These deep learning experiments were run on Linux architecture.

Table 3.3: The experiments which were run to obtain my results where the first column is the training data and the second column is the testing data.

Training Data	**Testing Data**
English	English
French	English
Russian	English
French, Russian	English
English	French
French	French
Russian	French
English, Russian	French
English	Russian
French	Russian
Russian	Russian
English, French	Russian

The first value in Table 3.3 in each row is the training data, while the second value

indicates the testing data. This tests the model's performance in a monolingual setting and a multilingual setting. By these experiments, I also gain insight into how the models perform when given two languages as training data and testing on one held out language while still maintaining the cross-lingual zero-shot setting. By design, these experiments are also performed in a zero-shot setting. In this setting, zero-shot refers to the model never seeing a language before testing. Typically, zero-shot refers to the model predicting classes it has not seen in training [3]. In spam detection, this could be seen as training on only spam emails and seeing if the model can still predict if an email is ham. However, there are different forms of zero-shot and in my case, it refers to the model being asked to make predictions on a language it has not previously seen, while having seen all of the classes during training, both spam and ham which is the definition used by Schwenk and Li [75]. Being performed in a zero-shot setting is important as many languages are considered low-resource languages and have little to no email data available to researchers. If a model can correctly classify emails as spam or ham in a language it has never seen before, by being trained on another language or languages that I do have available, this would be a significant accomplishment in the field of phishing detection. With this, end users who speak a low resource language such as Lithuanian can still be protected by a phishing detection system, and therefore, the safety of the system will be increased.

3.3.2 Downsampling Experiments

The downsampling experiments were performed to better understand the increase in accuracy when I train on two languages. The first possibility is that it is simply because I add more data to the model when I add another language, and the additional data increases my accuracy. The second possibility is that the extra diversity in the training data is the reason why I observe an increase to my accuracy when training on two languages. If I downsample the data, and the accuracy still is higher

or similar to the full data experiment with two languages, then it would suggest that diversity is a bigger factor than volume. If when downsampled, the accuracy is lower than the full data experiment with two languages, but higher than the experiment with one language, then it would suggest that diversity and volume both influence the result. Finally, if it is lower than both full data experiments, one language and two, then it suggests that volume is a bigger factor than diversity.

I also consider downsampled experiments using XLM RoBERTa. XLM-RoBERTa is considered, as during my experimental setup, this is the model that showed the most promising results in preliminary tests. First, I considered two scenarios for the downsampling experiments. In both setups, I trained on English and French and tested on Russian.

In the first downsampling experiment setup, I first obtain the downsampled data using the sample method in the pandas library in Python [88, 58] without setting the random state to a fixed value, only setting the number of emails to the size of the English training data, which is the larger of the two training datasets between English and French. The XLM RoBERTa model is subsequently trained and then tested. This process was repeated ten times, and the accuracy of the model was recorded with each iteration. After the experiments were run, the average (mean) and standard deviation were calculated using these values.

There exists a limitation in this experimental setup which is that the data used is highly dependent on the aforementioned sample method. The reasoning behind not setting the random state is that I wish to explore different shuffles of the data to negate the possibility of one shuffle being better than another and simply getting fortuitous results due to a good shuffle that I chose beforehand by explicitly setting the random state to a fixed value. If I had set the state to a fixed value, there may exist a better random state which leads to better results. In this sampling methodology, the ratio of English emails to French emails, the ratio of English ham

to English spam, and the ratio of French ham to French spam could vary greatly between runs. For example, one run could consist of mostly English ham with only a very small amount of English spam and French spam or ham. This is, of course, a worst-case scenario and fairly unlikely to actually happen in practice, but demonstrates the limitation with the experiment that could happen, or happen to a lesser degree which still affects the experiments negatively.

The following experiment is ran in order to compare the data of the above experiment with the data obtained with this one. In this experiment, I first obtain the data by calling the sample method, selecting all the data, and shuffling it. In this experiment, only the order of the data will be different between runs, not the aforementioned ratios between languages and the spam to ham ratio. After this, as in the previous experiment, I train and test the XLM-RoBERTa model ten times, recording the accuracy each time as well. Once this was complete, the average (mean) and standard deviation were calculated.

From the two experiments I ran, I can then analyse if downsampling on average lessens my accuracy indicated by the mean and if downsampling means I have more disperse values indicated by the standard deviation. This will also serve to indicate whether the limitation listed above comes into effect when the ratios are not maintained. If the standard deviations are similar, it would suggest that even if I downsample and even if I get poor shuffles and splits in the data, the range of my values compared to only changing the order of the data is not affected.

Finally, I also conducted a controlled downsampling experiment that addresses a limitation of the experiment listed above. This experiment was where the ratio of language-to-language and ham-to-spam for each language was balanced to be the same as the full-scale experiment where I train on all of the data I have available in two languages.

In this experiment, Table 3.4 provides the amount of emails for each class and lan-

guage, and shows the ratio between languages, and classes for each language.

Table 3.4: Dataset details for the controlled downsampling experiment.

Experiment	**Language**	**Class**	**Count**
English & French	English (89%)	Ham (71%)	2517
		Spam (29%)	1028
	French (11%)	Ham (50%)	219
		Spam (50%)	219
English & Russian	English (96%)	Ham (71%)	2715
		Spam (29%)	1109
	Russian (4%)	Ham (50%)	80
		Spam (50%)	79
French & Russian	French (73%)	Ham (50%)	173
		Spam (50%)	172
	Russian (27%)	Ham (50%)	64
		Spam (50%)	63

From this, I maintain the ratios of the languages as well as the ham-to-spam ratios that exist in the full-size dataset. This experiment is only run once and is compared to the data obtained from running the experiments at full volume. For example, the downsampling experiment where I train on English and French and downsample to the size of English training, can be compared to the numbers I got when training on English and French together, as well as training on English by itself. By downsampling, I can see whether the addition of another language is beneficial to the training process. In addition, it will tell me whether any increase in performance when training on two languages is due to volume or diversity.

Chapter 4

Results

4.1 Overview

This chapter overviews the results of the experiments which are detailed in Chapter 3 where the methodology is explained. First, I detail the motivation behind the experiments and explain the metric used for evaluation including what baseline is used and how this affects the interpretation of our results. Following this, each subsection aims to answer a specific question; these questions include which models perform best on monolingual phishing detection (section 4.2), which models perform best in the cross-lingual zero-shot setting (section 4.3), how GPT performs at multilingual phishing detection (section 4.4), does training on multiple languages at once improve the performance (section 4.5), how downsampling affects the performance (section 4.6), and does diversity or volume of training data cause increases in our accuracy (Section 4.7).

The first research question, where I examine if transformer models are better than traditional machine learning models at monolingual detection is answered in section 4.2; question two, where I examine if transformer models are better than traditional models at cross-lingual zero-shot detection, is answered in section 4.3. The third

question about the performance of GPT compared to XLM-RoBERTa (XLMR) is answered in section 4.4. Finally, section 4.5 answers the final question of whether adding an additional language increases my performance. Sections 4.6 and 4.7 give additional insights into the final question.

4.1.1 Metric Used

My experiments use accuracy as the metric for evaluation. Accuracy is defined by the below formula where TP and TN are true positives and true negatives and FP and FN are false positives and false negatives:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

In simpler terms, accuracy is the metric of how many predictions the model gave correctly out of the total number of predictions.

Two out of three of my datasets, French and Russian, are balanced at 50% each, which means that accuracy is a good metric that is easy to understand. In the case of the English data, the classes are not balanced which means accuracy can be misleading. Accuracy can be misleading in this case because the model can simply pick one class every time and still achieve a high accuracy score. Although I could use precision and recall to remedy this issue, I instead interpret the results keeping in mind the baseline, which is the most frequent class baseline. The most frequent class baseline is determined by the model only picking one label, which is the label it saw most during training. We can shift our expectations when interpreting the English results in this way. Instead of the accuracy needing to surpass 50%, it needs to surpass the percentage of ham emails it saw in training as ham emails are seen far more than spam. In this case, it will be 72%. If a model does not surpass the baseline, this indicates that it learned poorly and can suggest that it is not a good

model. In a binary classification problem such as spam detection, the worst a model can do is actually meet the baseline (or perform around it) as opposed to scoring 0%. This is because 0% would mean the model learned everything backwards and if the labels were flipped, it would achieve 100%. However, if the model scores at or around the baseline, flipping the labels will not help as it would appear the model selected randomly or consistently selected the most frequent class and did not learn at all. Surpassing the baseline (in this case 50% for French and Russian and 72% for English) does not necessarily indicate the model performed well in comparison to other models; however, it indicates that the model learned something instead of simply selecting a random value or the most frequent class during training.

Table 4.1 shows my results for the tests in each language. Each row represents the model which was trained and tested for the experiment where the last row is the most frequent class baseline. This means that the model only predicts the class it saw the most during training. As stated above, I use this as a comparison especially for the English testing data, which is not evenly split across classes. For each column in the aforementioned tables, it represents the language or languages on which the models were trained. Each language is represented using its ISO 639-1 code, where en is English, fr is French, and ru is Russian.

4.2 Performance Analysis of Monolingual Phishing Detection Models

The results in Table 4.1 suggest that monolingual detection is largely a solved problem, which means that when a model trains and tests on the same language, the best models can detect phishing emails with nearly 100% accuracy or in the best cases — 100%. Even very basic traditional non-deep learning models such as logistic regression and SVM perform extremely well at monolingual detection with one

Table 4.1: Accuracy where the testing language is shown in the testing language column for each model. The bolded value is the best accuracy for each training language training on the indicated languages.

Testing Language	Model	en	fr	ru	fr,ru	en,ru	en,fr
	GPT-3	**0.99**	0.67	0.28	0.76		
	Logistic Regression	0.93	0.62	0.4	0.74		
	Random Forest	0.91	0.64	0.28	0.67		
en	SVM	0.78	**0.73**	0.45	0.75		
	XLM Roberta	**0.99**	0.72	**0.71**	**0.99**		
	GPT-2	**0.99**	0.72	0.66	0.61		
	Most Frequent Class	0.71	0.71	**0.71**	0.71		
	GPT-3	0.78	**1.00**	0.63		0.68	
	Logistic Regression	**0.79**	0.96	0.45		0.50	
	Random Forest	0.41	0.94	0.43		0.41	
fr	SVM	0.62	0.88	0.49		0.63	
	XLM Roberta	0.68	0.98	**0.68**		**0.99**	
	GPT-2	0.50	**1.00**	0.56		0.48	
	Most Frequent Class	0.50	0.50	0.50		0.50	
	GPT-3	0.81	**0.77**	**1.00**			0.77
	Logistic Regression	0.50	0.36	0.50			0.59
	Random Forest	0.54	0.45	0.50			0.50
ru	SVM	0.47	0.52	0.50			0.63
	XLM Roberta	**0.95**	0.50	0.97			**0.95**
	GPT-2	0.70	0.54	0.95			0.68
	Most Frequent Class	0.50	0.50	0.50			0.50

exception. In the case of Russian shown in Table 4.1, logistic regression, random forest, and SVM did not learn anything and therefore, outputted the most frequent class in all cases, achieving an accuracy of 50%. However, logistic regression scored 93% in the English test, only 6% lower than the leading results scored by the deep learning models of 99%. For French, logistic regression scored 96%, only 4% lower than the highest value scored by GPT-2 and GPT-3 at 100%. 4% and 6% are not an incredibly large drop in performance. Ultimately, high accuracy values are expected, and not remarkable, especially for deep learning models such as GPT-3 and XLM-RoBERTa. Previous research has shown exceptional results in this area, achieving the same results as I have—near or at 100% accuracy [35, 9, 24, 23, 44].

However, just because the results are not surprising does not mean that they do not

have value. These results suggest that as long as a system is trained in the language of its target audience, it will be highly likely to protect them from phishing emails. The caveat of this is that my experimental setup only uses the text from the body of the email for the model to make a classification on whether it is spam or ham. Phishing could be done solely through the usage of the subject line of the email as well as through an attachment. In this case, since the body would be blank, my model would not be able to accurately make a prediction. The other case is that the body of an email could contain an image for the scam as well. My system would not be able to extract the text of the image in its current state and make a prediction accurately as it only considers the text in the body of the email. Another caveat is that the dataset that I used for training, especially in the English case, is quite old, first created in 2004. For the French and Russian data, I do not have an age for the spam emails but the ham emails could be outdated as well in these cases, with the data coming from datasets made in 2004 and 2007. It is highly possible that phishing emails have evolved since 2004 and 2007 and ham emails would have also as language evolves. While the training data for my system in English is very business focused, there has been research done into the field of domain adaptation for phishing detection, meaning that even a ham email that is not business focused and is written using modern language could still be detected as ham[4, 28, 64, 67].

Large language models, such as GPT-3, require an extremely large amount of training data. For my experiment, I am simply fine-tuning the model to my task. However, previously these models were trained on a very large amount of data. GPT-3 used 570GB of data after filtering in its training[16], while XLM RoBERTa used 2.5TB of data [21].

The other consideration is environmental costs when considering traditional methods of machine learning over deep learning methods, including the newest large-language models. GPT-3 is estimated to have produced 578,460 KG of carbon dioxide in

emissions on a single training run on North American servers, as shown by Taddeo et al. [81]. However, the authors note that geography is important, considering that if the training is done on South African servers, the estimate jumps significantly to 942 330 KG in CO2 emissions. Currently, the exact amount of emissions and exactly what servers the training was completed on have not been disclosed by OpenAI. Other research supports the findings that large language models can have a significant negative effect on the environment in terms of emissions [11, 87, 7, 68].

Finally, I can also consider the computation and training time. To fine-tune a pre-trained deep learning model, the performance varies greatly based on the hardware. If the training is done on a CPU compared to a GPU, it will have a very large impact on training time. However, even with good hardware and powerful GPUs, it is not uncommon for a pre-trained deep learning model to take hours or days to finish fine tuning. Although not explicitly captured, as it is not the goal of my research to compare, training XLM RoBERTa on GPUs such as the NVIDIA Quadro P5000, the NVIDIA Quadro RTX 5000, and the RTX A4000, training often took more or less an hour with a batch size of 16. Compared to my feature-based methods using traditional machine learning methods, this is a massive increase in time. In the largest monolingual case where the all three models are trained on English and tested on English, each model is trained and tested in under 5 seconds using only a CPU. This is a massive decrease in time.

With all of this being said, since monolingual detection appears to be a largely solved problem, I show that traditional methods perform slightly weaker than deep learning models. However, at the slight cost of accuracy, I see other benefits in the form of no huge amount of data is needed to train traditional models beforehand, they are better for the environment, and they run and train much faster.

4.3 Performance Analysis of Cross-Lingual Zero-Shot Phishing Detection Models

The results largely show that cross-lingual detection while training on a single language is a very difficult problem with which models still struggle. Traditional feature-based methods, using non-deep learning models, often perform underneath the baseline. In addition, when they do perform above the baseline, it is often not by a considerable amount especially when compared to my deep learning results. The two exceptions are the tests in Table 4.1 where SVM performs the best when training on French and testing on English and logistic regression performs the best when training on English and testing on French. I can also say that SVM trained on English and tested on French performs well, outperforming the baseline by 12 percentage points and only losing to XLM-RoBERTa by 6 percentage points. With this being said, it still loses to logistic regression by 17 percentage points and GPT-3 by 16 percentage points, making it not a good result by comparison.

While zero-shot cross-lingual classification has proven to be challenging, XLM-RoBERTa still performs remarkably well. Overall, XLM-RoBERTa performed the best out of any of the models even if it has some tests it performs poorly at such as the test in which it trains on Russian and tests on English which performs on par with the baseline, as highlighted in Table 4.1. It is currently unknown why it performs so poorly in this setting and could be explored further in future work. Out of the 9 cross-lingual experiments performed, XLM-RoBERTa achieves the best score in 6 of these tests, which means that it performs the best in 66% of the experiments. So, overall, XLM-RoBERTa can be viewed as the best model for cross-lingual zero-shot phishing detection.

This is not to say that the GPT series does not perform well also, it is just to say that it is does not perform as well and, in some cases, it performs quite poorly. The

performance of the GPT series specifically will be discussed in the following section.

4.4 Performance Analysis of a Novel Large Language Model (GPT)

I also show that GPT 2 and 3 have limitations and, while they excel at generating text, can perform poorly compared to XLM-RoBERTa. This is important to note as there is a substantial amount of research that shows that GPT performs many complicated tasks, such as passing the bar exam in the USA [14, 41] and passing a medical exam in Japan[40]. There is a massive push by industry to include new large language models in every product with Google recently embedding AI into 25 products [29]. With such a large push to use these powerful models, it is important to determine whether they do indeed have limitations, and in which circumstances end users should not depend on them to be able to perform tasks reliably. In terms of this research, overall GPT models do not perform multilingual phishing detection as well as other models, such as XLM RoBERTa.

One reason for this is that XLM-RoBERTa was explicitly trained to be multilingual [21], while GPT-3 was not [16], although some of the training data for GPT was in other languages besides English and as a result it does have some multilingual capacity. This finding should also extend to the current ChatGPT as well because it trained on similar data.

Based on GPT-4's technical report however, it could perform well in this instance. In this report, OpenAI notes its performance in MMLU (Massive Multitask Language Understanding) tasks for numerous languages, including low-resource languages such as Nepali, Urdu, and Ukrainian [56]. However, this experiment is done in a three-shot setting, while my research focuses on zero-shot settings. While they are making a point to include tests for numerous and low resource languages, it may still not

perform as well as a dedicated multilingual model. Future work can include this area and see how GPT-4 performs in this setting as it is slowly being rolled out for researchers and the public.

4.5 Analysis of Training on Multiple Languages

My results also show that, simply by adding an additional language to training, I can increase the accuracy of my models on average. This is evident in Table 4.1 (see page 41) where XLM-RoBERTa performs better when trained on two languages rather than either of them separately. I also see this property in the case of the traditional models logistic regression, random forest, and SVM, that simply by training on two languages combined, I can improve the accuracy over training on one alone. For the previous four models, my findings show that the increase in accuracy when adding an additional language to training is consistent across most experiments.

This does come with a caveat however, which is that from these results alone, it is unclear if the increase in accuracy comes from the addition of another language to the training data or if it simply is because of more data being added. This is examined in the following sections.

However, on their own, some of the results are very promising, such as the experiment in Table 4.1 where XLM-RoBERTa is trained on both Russian and French and tested on English. Training on French on its own only achieves 72% while training on Russian only achieves 71%. Taking into account that the baseline is 71%, these are poor results. It means for the most part that the model did not learn anything and was unable to properly predict an instance it receives, so it defaults to the most frequent class baseline. However, when I train on the combination of both languages, I achieve 99% indicating that the model learns very effectively. It would be a great advancement if models were able to have a large improvement in accuracy simply by

adding another language. This is especially true for low-resource languages. A low enough resource language is expected to be an exception to the previous claim that monolingual detection is a largely solved problem. Since they are low resource and there is limited training data for them, it is expected that a model would struggle with detecting spam in a low enough resource language due to the lack of training data since it would have insignificant examples from which to learn the distinctions between the two classes of spam and ham. It would be beneficial if I could simply train on English data, which is plentiful, or English combined with another language that is available in order to achieve even better results. In the following section, where I perform the downsampling experiments, I am also simulating low-resource languages as I ignore training data.

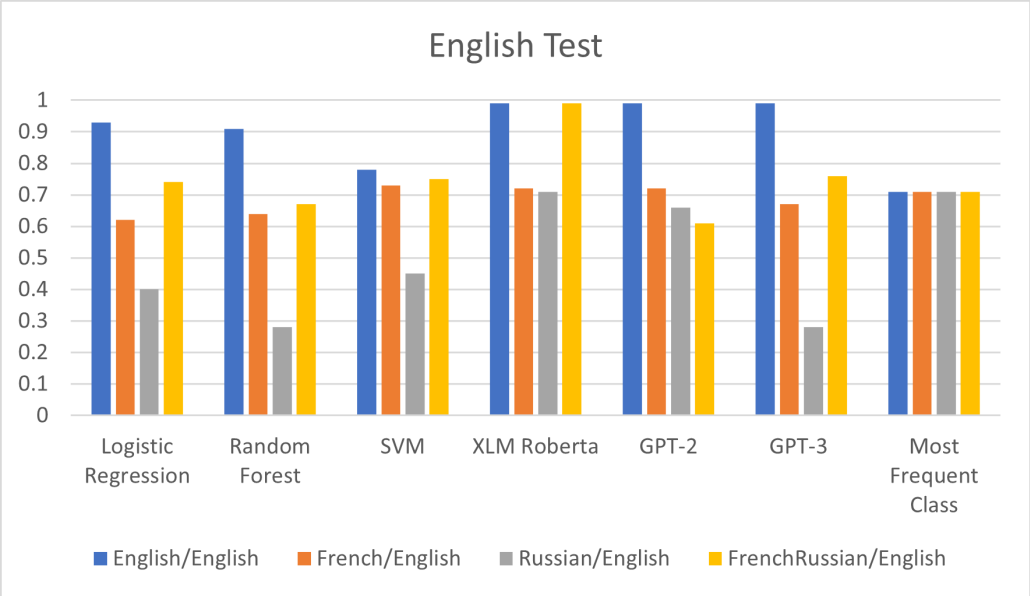

Figure 4.1: A visual representation of a portion of the data where English is the testing language in Table 4.1 which shows the results in accuracy when I train on French and Russian and test on English.

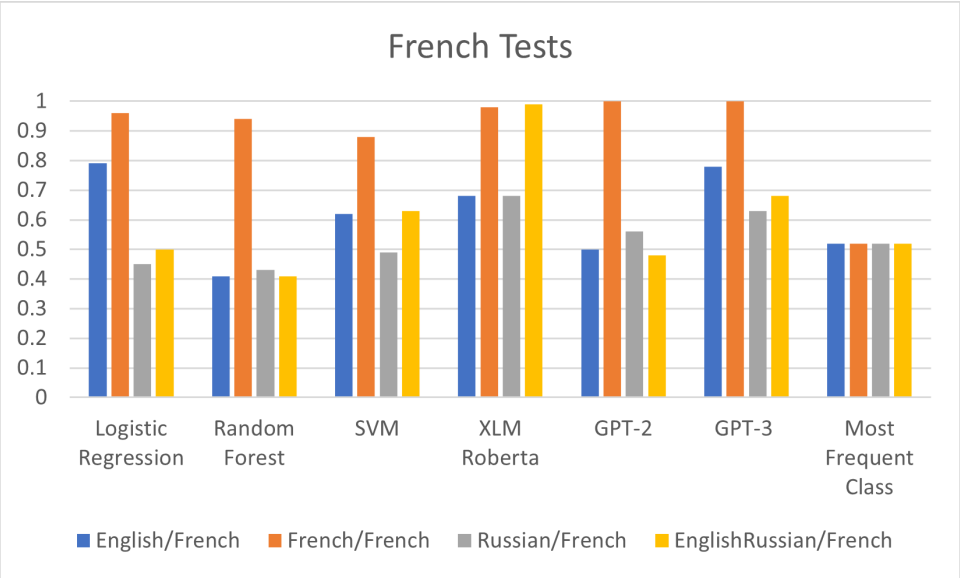

Figure 4.2: A visual representation of a portion of the data in Table 4.1 which shows the results in accuracy when I train on English and Russian and test on French.

4.6 Analysis of Downsampling on my Results

As detailed in Chapter 3.3, I run two experiments in which I examine the results in terms of accuracy from training XLM-RoBERTa on English and French and testing on Russian. In one of these experiments, I only use the size of the English data in training, and in one, I use all available data. The size of the English data is used as this is the larger of the two languages between English and French. I consider the larger of the two as it will allow the model to have more training data. The experiment in which I use all the data is replicating the setup in the previous experiments in which I train on two languages and is used for a point of comparison.

With the number of training instances equal to 5570 (the full number of instances), I obtain an average (mean) result of 89.5 and a standard deviation of 7.39 which is shown in Figure 4.4.

When I train on 3983 emails (the size of the English training dataset), I obtain an average of 83.6 and a standard deviation of 7.36 as also shown in Figure 4.4.

These results, summarized in Figure 4.4, show that I do indeed obtain better results

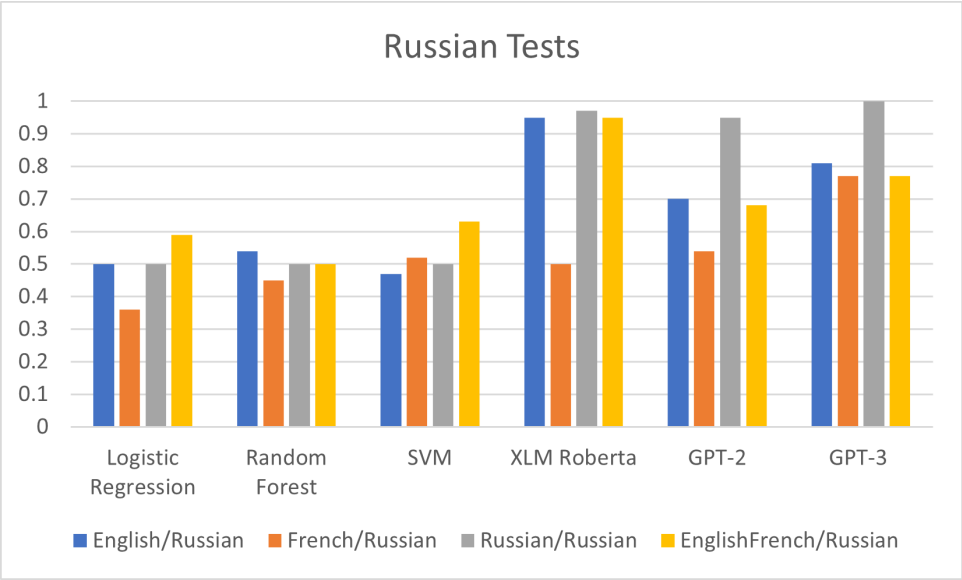

Figure 4.3: A visual representation of a portion of the data in Table 4.1 which shows the results in accuracy when I train on English and French and test on Russian.

when I use all of the available training data. This is not surprising. It has been shown [86, 39, 16] that the inclusion of additional training data is directly linked to an increase in performance for deep learning and transformer models, so the fact that my model exhibits the same behaviour is not unexpected. What is valuable to examine is the problem of whether my model only has an increase in accuracy due to the increase in training data or whether it is due to the addition of another language during training. As mentioned above in Chapter 4.3, 4 models all exhibit this property of having an increase in accuracy when an additional training language is added, so whether the increase is because of added volume or diversity is important to discuss as well. This question is examined in Chapter 4.5.

The downsampling experiment detailed in this section uses English and French for its training languages and Russian as its testing language but I could also run these on other language pairs to confirm my results. However, with the data and results which have been obtained already, it is believed that these would not help to answer the question of volume versus diversity, as trends in the previous experiments hold true for each language suggesting it could be similar in this case, and therefore, performing

additional experiments with the other language pairs is left to be explored in future work.

I can also examine the standard deviation obtained from the experiments. By examining the standard deviation, I can clearly see a negligible difference in the standard deviation between the two experiments. This indicates that the results were not spread out much farther than the average for one experiment over another. One might think that the downsample experiment would have more of a spread and varying results due to the aforementioned limitation of varying ratios between training languages and classes; however, my results show otherwise. From these results, the standard deviations of both experiments suggest that the variation was not large enough to greatly impact the experiment and the accompanying results. In the following subsection, the experiment addresses the limitation between ratios of classes and languages.

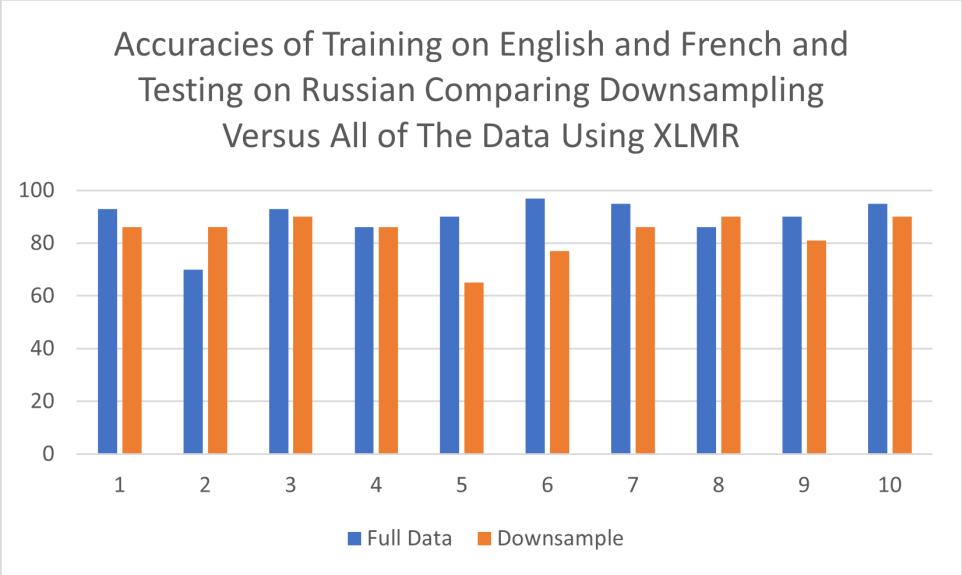

Figure 4.4: Downsample experiment comparing English and French using XLM-RoBERTa over 10 runs.

4.7 Analysis of Diversity Versus Amount of Training Data in Two-Language Training Experiments

As discussed in Chapter 3.3, I run three experiments to determine whether the increase in accuracy I saw when adding another language was because of diversity in the data or simply an increase in volume of the data.

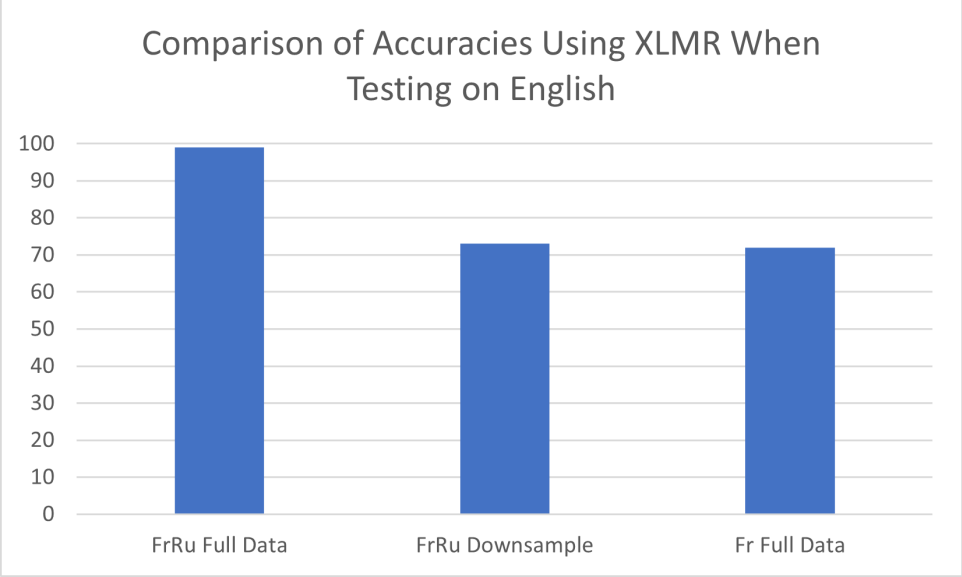

Figure 4.5: Downsample experiment comparing training on all of the data over French and Russian, the downsampled data over French and Russian, and all of the French data.

4.7.1 Experiment One

Case one, where I train XLM-RoBERTa on French and Russian and test on English shows that when I downsample, I see a significant decrease in performance compared to when I use all of the data. Furthermore, I can see that, compared to the test where I train on French alone, it is just about equal as shown in Figure 4.5. In this case, the results do not show, and are unclear about, whether the increase was due to an

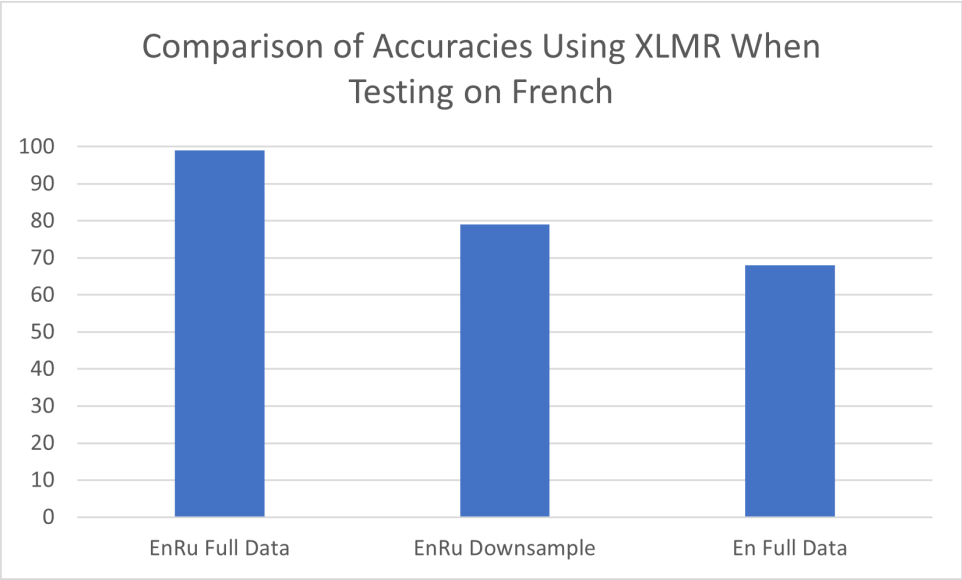

Figure 4.6: Downsample experiment comparing training on all of the data over English and Russian, the downsampled data over English and Russian, and all of the English data.

increased amount of data or increased diversity in the data.

4.7.2 Experiment Two

The second case where I train XLM-RoBERTa on English and Russian and test on French shows that when I downsample, I have an improvement still over training on the English data alone which is shown in 4.6. However, it is still a decrease over training on all of the English and Russian data. This case suggests that training on multiple languages improves performance, ie., that an increase in diversity of training languages contributes to an increase in performance. This is because I increase my accuracy over training on English alone, and since I have the same amount of training instances, it stands to reason that the addition of Russian is what causes the increase. The reason that volume is still beneficial is that the model still does not perform as well when the downsampling occurs compared to using all of the data. This suggests that the amount of data used is important, as suggested in the literature [86, 16, 39].

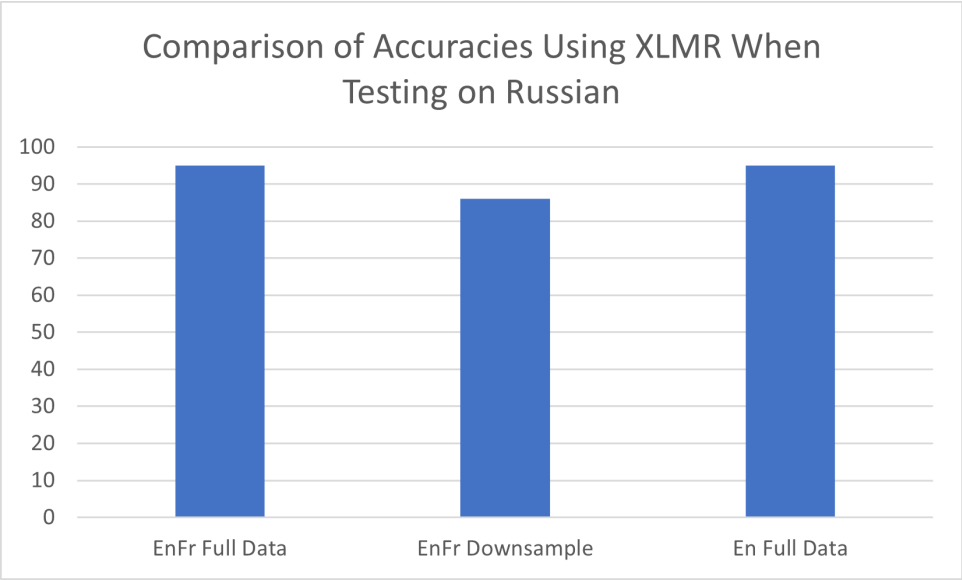

Figure 4.7: Downsample experiment comparing training on all of the data over English and French, the downsampled data over English and French, and all of the English data.

4.7.3 Experiment Three

Finally, in the third case, which is when I train on English and French and test on Russian, the results show a massive decrease in accuracy compared to both when I train on English and French at full volume and when I train on English alone as displayed in Figure 4.7. This is in stark contrast to my previous case, as it suggests that diversity of training languages is actually harmful to accuracy. Comparing the results when I downsample and the results training only on English, my downsample experiment performs relatively poorly, which indicates that it is the addition of French into the training dataset which causes the decrease, as the number of instances is exactly the same.

4.7.4 Results of the Diversity Versus Amount of Training Data in Two-Language Training Experiments

In my three cases, I demonstrate that the results are overall unclear. I am not sure whether the increase in accuracy when I train on two languages which is shown in Figures 4.1, 4.2, and 4.3 is because the models were trained on another language or whether it is because the models were simply trained on more data. However, this experiment also had a limitation in that the controlled downsample experiment was run only once. If each downsample experiment was run many times (eg, 10 times) it would mitigate the risk that the findings are an artefact of a particular shuffle of the data. By doing so, an average could also be taken which would be more accurate. This could show different results than those I have listed above and answer the question of diversity compared to volume.

However, it is highly likely, and the results support it, that volume has a positive effect on accuracy. This is to say that when more data is added, my accuracy increases. In the downsample experiments, I see this very clearly. Figures 4.5, 4.6, and 4.7 all show that when I downsample, I lose accuracy compared to when I use all of my data. These results suggest that, as said above, volume is important to the performance of the model. This is a well known finding[86, 16, 39].

Chapter 5

Conclusion

While phishing is a large problem, we show that there are ways that we can combat this growing issue. By leveraging pretrained multilingual models such as XLMR, we can achieve good performance detecting phishing emails in a language not seen during training. This is important because systems can start to learn to detect what phishing emails look like in numerous languages, even though training data for that language may not exist or exist in only a very small amount.

In this work, I explored four main research questions: whether transformer models perform better than traditional machine learning models at monolingual phishing detection, whether transformers models perform better than traditional machine learning models at cross-lingual zero-shot phishing detection, whether GPT-3 performs better than XLM-RoBERTa, and if training on two languages is better than training on one.

I show that monolingual detection is generally a solved problem, but there are a variety of considerations when choosing a model. In terms of accuracy, transformers perform better, but it comes with the cost of higher training times and an environmental cost in terms of C02 emissions compared to the traditional models which have less training time as well as less emissions. For cross-lingual zero-shot detection,

transformer models have a much higher accuracy than traditional models, offsetting the cost of time. When comparing GPT-3 and XLM-RoBERTa, XLM-RoBERTa performs better; however, it should be noted that GPT-3 performs remarkably well for not being explicitly trained to be multilingual. Finally, I demonstrate that by simply adding another language to the training set, I can improve my accuracy. However, whether this is due to adding more data or adding diversity to the data remains an interesting question for future work.

5.1 Future Research Directions

This section identifies interesting directions that future work may consider in the area of cross-lingual zero-shot phishing detection. These directions expand and enhance my work performed in this thesis, answer a new question in this field, and provide a direction which would greatly aid future research.

5.1.1 Addressing Limitations of This Research

One future direction includes addressing the limitations mentioned in Chapter 3. Although I address the limitation of the ratios of the data not being the same, the experiment which addresses it is only ran once, and only for one model. Therefore, this work would be enhanced by an experiment which addresses this limitation by testing multiple models, running the experiments 10 times each, and obtaining an average.

5.1.2 Few Shot Learning

Another direction that could be explored is few shot learning. Low resource languages may have a few emails available to the developers of a spam detection system that they may be able to use in the training of the model. As mentioned in my work, the

GPT 3 and 4 series were trained as, and ideally used as, few shot models[16, 56]. It could be that these models perform excellently in a multilingual setting for phishing detection when trained as few shot models.

5.1.3 Creation of a Multilingual Email Dataset

Finally, this work would be enhanced by the creation of a real non-translated multilingual email dataset that includes both spam emails and ham emails in multiple languages, and as such, is a consideration for a future direction as well.

Bibliography

[1] *2007 trec public spam corpus*, `https://plg.uwaterloo.ca/\~gvcormac/treccorpus07/`, 2007.

[2] Kayode S Adewole, Abimbola G Akintola, Shakirat A Salihu, Nasir Faruk, and Rasheed G Jimoh, *Hybrid rule-based model for phishing urls detection*, Emerging Technologies in Computing: Second International Conference, iCETiC 2019, London, UK, August 19–20, 2019, Proceedings 2, Springer, 2019, pp. 119–135.

[3] Kabir Ahuja, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury, *Multi task learning for zero shot performance prediction of multilingual models*, https://arxiv.org/abs/2205.06130, 2022.

[4] Mustafa Aydin and Nazife Baykal, *Feature extraction and classification phishing websites based on url*, 2015 IEEE Conference on Communications and Network Security (CNS), 2015, pp. 769–770.

[5] M. Tariq Banday and Shafiya Afzal Sheikh, *Multilingual e-mail classification using bayesian filtering and language translation*, 2014 International Conference on Contemporary Computing and Informatics (IC3I), 2014, pp. 696–701.

[6] Ram B Basnet, Andrew H Sung, and Quingzhong Liu, *Rule-based phishing attack detection*, Proceedings of the International Conference on Security and Management (SAM), Citeseer, 2011, p. 1.

[7] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell, *On the dangers of stochastic parrots: Can language models be too big?*, Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, 2021, pp. 610–623.

[8] *Beware of fraudulent emails appearing to come from service-public.fr!*, `https://www.service-public.fr/particuliers/actualites/A14806?lang=en`, April 2021, Accessed June 20, 2022.

[9] Uma Bhardwaj and Priti Sharma, *Email spam detection using bagging and boosting of machine learning classifiers*, International Journal of Advanced Intelligence Paradigms **24** (2023), no. 1-2, 229–253.

[10] BigScience Workshop, *Bloom (revision 4ab0472)*, 2022.

[11] Abeba Birhane, Atoosa Kasirzadeh, David Leslie, and Sandra Wachter, *Science in the age of large language models*, Nature Reviews Physics (2023), 1–4.

[12] Ivan Blagojević, *Email statistics 2022*, `https://99firms.com/blog/how-many-email-users-are-there/\#gref`, Accessed Aug 16, 2022.

[13] *Blocked attachments in outlook*, https://support.microsoft.com/en-us/office/blocked-attachments-in-outlook-434752e1-02d3-4e90-9124-8b81e49a8519, Accessed Sept 13, 2022.

[14] Michael Bommarito II and Daniel Martin Katz, *Gpt takes the bar exam*, arXiv preprint arXiv:2212.14402 (2022).

[15] Hadj Ahmed Bouarara, Reda Mohamed Hamou, and Abdelmalek Amine, *A novel bio-inspired approach for multilingual spam filtering*, Int. J. Intell. Inf. Technol. **11** (2015), no. 3, 45–87.

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, *Language models are few-shot learners*, CoRR **abs/2005.14165** (2020).

[17] *Business email compromise: The $43 billion scam*, `https://www.ic3.gov/Media/Y2022/PSA220504`, May 2022, Accessed June 6, 2022.

[18] Yong Cheng, Yang Liu, Tianjian Chen, and Qiang Yang, *Federated learning for privacy-preserving ai*, Commun. ACM **63** (2020), no. 12, 33–36.

[19] Federal Trade Commission, *Phishing scams and how to spot them*, https://www.ftc.gov/news-events/topics/identity-theft/phishing-scams, Jul 2021.

[20] Wm Arthur Conklin, Gregory B. White, Chuck Cothren, Roger Davis, and Dwayne Williams, *Comptia security+ exam guide, (exam sy0-501)*, McGraw-Hill Education, 2018.

[21] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov, *Unsupervised cross-lingual representation learning at scale*, CoRR **abs/1911.02116** (2019).

[22] *E-mail format*, `https://www.geeksforgeeks.org/e-mail-format/`, Accessed Sept 13, 2022.

[23] Maurice Aniefiok Ebong, *Deep learning phishing email classifier combined with nlp*, Ph.D. thesis, Dublin, National College of Ireland, 2022.

[24] Nusrat Jahan Euna, Syed Md Minhaz Hossain, Md Musfique Anwar, and Iqbal H Sarker, *Content-based spam email detection using an n-gram machine learning approach*, Applied Intelligence for Industry 4.0, Chapman and Hall/CRC, 2023, pp. 176–187.

[25] Yong Fang, Cheng Zhang, Cheng Huang, Liang Liu, and Yue Yang, *Phishing email detection using improved rcnn model with multilevel vectors and attention mechanism*, IEEE Access **7** (2019), 56329–56340.

[26] *Federal bureau of investigation internet crime report 2021*, `https://www.ic3.gov/Media/PDF/AnnualReport/2021\_IC3Report.pdf`, May 2022, Accessed June 6, 2022.

[27] Canadian Centre for Cyber Security, *Don't take the bait: Recognize and avoid phishing attacks - itsap.00.101*, https://www.cyber.gc.ca/en/guidance/dont-take-bait-recognize-and-avoid-phishing-attacks, Aug 2022.

[28] Aryya Gangopadhyay, Iyanuoluwa Odebode, and Yelena Yesha, *A domain adaptation technique for deep learning in cybersecurity*, On the Move to Meaningful Internet Systems: OTM 2019 Workshops: Confederated International Workshops: EI2N, FBM, ICSP, Meta4eS and SIAnA 2019, Rhodes, Greece, October 21–25, 2019, Revised Selected Papers, Springer, 2020, pp. 221–228.

[29] Nico Grant, *Google builds on tech's latest craze with its own a.i. products*, The New York Times (2023).

[30] Albert Harris and Dave Yates, *Phishing attacks over time: a longitudinal study*, (2015).

[31] Wadi' Hijawi, Hossam Faris, Ja'far Alqatawna, Ala' M. Al-Zoubi, and Ibrahim Aljarah, *Improving email spam detection using content based feature engineering approach*, 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2017, pp. 1–6.

[32] Wadi' Hijawi, Hossam Faris, Ja'far Alqatawna, Ibrahim Aljarah, Ala'M Al-Zoubi, and Maria Habib, *Emfet: E-mail features extraction tool*, 2017.

[33] Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural computation **9** (1997), no. 8, 1735–1780.

[34] *How to recognize and avoid phishing scams*, `https://consumer.ftc.gov/articles/how-recognize-and-avoid-phishing-scams`, Accessed Sept 13, 2022.

[35] Rashmi Jha and Gaurav Kunwar, *Machine learning based url analysis for phishing detection*, 2023 6th International Conference on Information Systems and Computer Networks (ISCON), IEEE, 2023, pp. 1–5.

[36] Sanaa Kaddoura, Omar Alfandi, and Nadia Dahmani, *A spam email detection mechanism for english language text emails using deep learning approach*, 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE, 2020, pp. 193–198.

[37] Leslie Kaelbling, *Enron email dataset*, `https://www.cs.cmu.edu/~enron/`, 2015.

[38] Micheal Kan, *Bogus porn emails using old passwords to scam you out of cash*, `https://www.pcmag.com/news/bogus-porn-emails-using-old-passwords-to-scam-you-out-of-cash`, 2018, Accessed Aug 20, 2022.

[39] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, *Scaling laws for neural language models*, CoRR **abs/2001.08361** (2020).

[40] Jungo Kasai, Yuhei Kasai, Keisuke Sakaguchi, Yutaro Yamada, and Dragomir Radev, *Evaluating gpt-4 and chatgpt on japanese medical licensing examinations*, arXiv preprint arXiv:2303.18027 (2023).

[41] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo, *Gpt-4 passes the bar exam*, Available at SSRN 4389233 (2023).

[42] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones, *Phishing detection: a literature survey*, IEEE Communications Surveys & Tutorials **15** (2013), no. 4, 2091–2121.

[43] Bryan Klimt and Yiming Yang, *Introducing the enron corpus.*, CEAS, vol. 45, 2004, pp. 92–96.

[44] Mehmet Korkmaz, Emre Koçyiğit, Özgür Koray Şahingöz, and Banu Diri, *A hybrid phishing detection system using deep learning-based url and content analysis*, (2022).

[45] Elmer Lastdrager, *Achieving a consensual definition of phishing based on a systematic review of the literature*, Crime Science **3** (2014), 9.

[46] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE **86** (1998), no. 11, 2278–2324.

[47] Younghoo Lee, Joshua Saxe, and Richard Harang, *Catbert: Context-aware tiny bert for detecting social engineering emails*, https://arxiv.org/abs/2010.03484, 2020.

[48] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692 (2019).

[49] Michael Lynn, *Scarcity effects on value: A quantitative review of the commodity theory literature*, Psychology & Marketing **8** (1991), no. 1, 43–57.

[50] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras, *Spam filtering with naive bayes-which naive bayes?*, CEAS, vol. 17, Mountain View, CA, 2006, pp. 28–69.

[51] Mahmood Moghimi and Ali Yazdian Varjani, *New rule-based phishing detection method*, Expert Systems with Applications **53** (2016), 231–242.

[52] Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava, *A survey on security and privacy of federated learning*, Future Generation Computer Systems **115** (2021), 619–640.

[53] Jose Nazario, *Index of / jose/phishing*, `https://monkey.org/~jose/phishing/`, 2016.

[54] Long Nguyen, Dinh Nguyen, Le Diep, Vu Tuan, Quang Anh Tran, and Bui Lam, *Detecting vietnamese spams using a multi-objective evolutionary approach*, (2017).

[55] Michael Nieles, Kelley Dempsey, and Victoria; Yan Pillitteri, *An introduction to information security*, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf, Jun 2017.

[56] OpenAI, *Gpt-4 technical report*, 2023.

[57] Duo Pan, Ellen Poplavska, Yichen Yu, Susan Strauss, and Shomir Wilson, *A multilingual comparison of email scams*, 2020.

[58] The pandas development team, *pandas-dev/pandas: Pandas*, https://doi.org/10.5281/zenodo.3509134, February 2020.

[59] Gilchan Park and Julia Rayz, *Ontological detection of phishing emails*, 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 2858–2863.

[60] Jeffrey Pennington, Richard Socher, and Christopher D Manning, *Glove: Global vectors for word representation*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[61] *Phishing attack statistics 2022*, `https://www.cybertalk.org/2022/03/30/top-15-phishing-attack-statistics-and-they-might-scare-you/`, Mar 2022, Accessed June 6, 2022.

[62] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, *Language models are unsupervised multitask learners*, (2019).

[63] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, Journal of Machine Learning Research **21** (2020), no. 140, 1–67.

[64] Alan Ramponi and Barbara Plank, *Neural unsupervised domain adaptation in nlp—a survey*, arXiv preprint arXiv:2006.00632 (2020).

[65] Justinas Rastenis, Simona Ramanauskaitė, Ivan Suzdalev, Kornelija Tunaityte, Justinas Janulevicius, and Antanas Cenys, *Multi-language spam/phishing classification by email body text: Toward automated security incident investigation*, Electronics **10** (2021), 668.

[66] Leonard Richardson, *Beautiful soup*, `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`, 2020.

[67] Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl, *Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification*, arXiv preprint arXiv:1908.11860 (2019).

[68] Matthias C Rillig, Marlene Ågerstrand, Mohan Bi, Kenneth A Gould, and Uli Sauerland, *Risks and benefits of large language models for the environment*, Environmental Science & Technology **57** (2023), no. 9, 3464–3466.

[69] David Ruano-Ordas, *Corpus 200 emails*, 2015.

[70] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, *Learning internal representations by error propagation*, Tech. report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[71] ———, *Learning representations by back-propagating errors*, nature **323** (1986), no. 6088, 533–536.

[72] Muborak Sagatova, *The relationship between a language and culture*, International Journal on Integrated Education **5** (2022), no. 1, 164–167.

[73] Said Salloum, Tarek Gaber, Sunil Vadera, and Khaled Shaalan, *Phishing email detection using natural language processing techniques: a literature survey*, Procedia Computer Science **189** (2021), 19–28.

[74] M SatheeshKumar, KG Srinivasagan, and G UnniKrishnan, *A lightweight and proactive rule-based incremental construction approach to detect phishing scam*, Information Technology and Management (2022), 1–28.

[75] Holger Schwenk and Xian Li, *A corpus for multilingual document classification in eight languages*, Proceedings of the Eleventh International Conference on

Language Resources and Evaluation (LREC 2018) (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[76] Zeeshan Bin Siddique, Mudassar Ali Khan, Ikram Ud Din, Ahmad Almogren, Irfan Mohiuddin, and Shah Nazir, *Machine learning-based detection of spam emails*, Scientific Programming **2021** (2021).

[77] Gunikhan Sonowal, *Phishing email detection based on binary search feature selection*, SN Computer Science **1** (2020), no. 4, 1–14.

[78] Spam Assassin Project, *Spamassassin public mail corpus*, https://spamassassin.apache.org/old/publiccorpus/, Accessed: 2021-11-29.

[79] *Spanish police warn public of correos phishing scam*, `https://www.healthplanspain.com/blog/spain-news/1461-spanish-police-warn-public-of-correos-phishing-scam.html`, April 2022, Accessed June 20, 2022.

[80] Yuwei Sun, Ng Chong, and Hideya Ochiai, *Privacy-preserving phishing email detection based on federated learning and lstm*, arXiv preprint arXiv:2110.06025 (2021).

[81] Mariarosaria Taddeo, Andreas Tsamados, Josh Cowls, and Luciano Floridi, *Artificial intelligence and the climate emergency: Opportunities, challenges, and recommendations*, One Earth **4** (2021), no. 6, 776–779.

[82] Chandra Thapa, Jun Tang, Sharif Abuadbba, Yansong Gao, Yifeng Zheng, Seyit Ahmet Çamtepe, Surya Nepal, and Mahathir Almashor, *Fedemail: Performance measurement of privacy-friendly phishing detection enabled by federated learning*, ArXiv **abs/2007.13300** (2020).

[83] Chandra Thapa, Jun Wen Tang, Alsharif Abuadbba, Yansong Gao, Seyit Camtepe, Surya Nepal, Mahathir Almashor, and Yifeng Zheng, *Evaluation of*

federated learning in phishing email detection, arXiv preprint arXiv:2007.13300 (2020).

[84] Jörg Tiedemann and Santhosh Thottingal, *OPUS-MT – building open translation services for the world*, Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (Lisboa, Portugal), European Association for Machine Translation, November 2020, pp. 479–480.

[85] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom, *Llama 2: Open foundation and fine-tuned chat models*, 2023.

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, 2017.

[87] Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa

Kasirzadeh, et al., *Taxonomy of risks posed by language models*, 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 214–229.

[88] Wes McKinney, *Data Structures for Statistical Computing in Python*, Proceedings of the 9th Python in Science Conference (Stéfan van der Walt and Jarrod Millman, eds.), 2010, pp. 56 – 61.

[89] *What is a proper email structure?*, `https://www.interserver.net/tips/kb/proper-email-structure/`, Accessed Sept 13, 2022.

[90] Tariku Yabshe, *Phishing email detection by using machine learning techniques*, Ph.D. thesis, St. Mary's University, 2022.

[91] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu, *Federated learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning **13** (2019), no. 3, 1–207.

[92] Adwan Yasin and Abdelmunem Abuhasan, *An intelligent classification model for phishing email detection*, arXiv preprint arXiv:1608.02196 (2016).

[93] Weiwei Zhuang, Qingshan Jiang, and Tengke Xiong, *An intelligent anti-phishing strategy model for phishing website detection*, 2012 32nd International Conference on Distributed Computing Systems Workshops, 2012, pp. 51–56.

Appendix A

Machine Learning Features

- **Email Body Feature Details**
  - ID: 1, Feature Details: Count of Spam Words
  - ID: 2, Feature Details: Count of Function Words
  - ID: 3, Feature Details: Count of HTML Anchor
  - ID: 4, Feature Details: Count of Unique HTML Anchor
  - ID: 5, Feature Details: Count of HTML Not Anchor
  - ID: 6, Feature Details: Count of HTML Image
  - ID: 7, Feature Details: Count of HTML All Tags
  - ID: 8, Feature Details: Count of Alpha-numeric Words
  - ID: 9, Feature Details: TF-ISF
  - ID: 10, Feature Details: TF·ISF without stopwords
  - ID: 11, Feature Details: Count of duplicate words
  - ID: 12, Feature Details: Minimum word length
  - ID: 13, Feature Details: Count of lowercase letters
  - ID: 14, Feature Details: Longest sequence of adjacent capital letters

- ID: 15, Feature Details: Count of lines
- ID: 16, Feature Details: Total number of digit characters
- ID: 17, Feature Details: Total number of whitespace characters
- ID: 18, Feature Details: Total number of uppercase characters
- ID: 19, Feature Details: Total number of characters
- ID: 20, Feature Details: Total number of tab characters
- ID: 21, Feature Details: Total number of special characters
- ID: 22, Feature Details: Total number of alphabetic characters
- ID: 23, Feature Details: Total number of words
- ID: 24, Feature Details: Average word length
- ID: 25, Feature Details: Words longer than 6 characters
- ID: 26, Feature Details: Total number of words (1–3 Characters)
- ID: 27, Feature Details: Number of single quotes
- ID: 28, Feature Details: Number of commas
- ID: 29, Feature Details: Number of periods
- ID: 30, Feature Details: Number of semicolons
- ID: 31, Feature Details: Number of question marks
- ID: 32, Feature Details: Number of multiple question marks
- ID: 33, Feature Details: Number of exclamation marks
- ID: 34, Feature Details: Number of multiple exclamation marks
- ID: 35, Feature Details: Number of colons
- ID: 36, Feature Details: Number of ellipsis
- ID: 37, Feature Details: Total number of sentences

- ID: 38, Feature Details: Total number of paragraphs
- ID: 39, Feature Details: Average number of sentences per paragraph
- ID: 40, Feature Details: Average number of words per paragraph
- ID: 41, Feature Details: Average number of characters per paragraph
- ID: 42, Feature Details: Average number of words per sentence
- ID: 43, Feature Details: Number of sentences beginning with uppercase
- ID: 44, Feature Details: Number of sentences beginning with lowercase
- ID: 45, Feature Details: Character frequency "$"
- ID: 46, Feature Details: Number of capitalized words
- ID: 47, Feature Details: Number of words in all uppercase
- ID: 48, Feature Details: Number of words that are digits
- ID: 49, Feature Details: Number of words containing only letters
- ID: 50, Feature Details: Number of words that are single letters
- ID: 51, Feature Details: Number of words that are single digits
- ID: 52, Feature Details: Number of words that are single characters
- ID: 53, Feature Details: Max ratio of uppercase letters to lowercase letters of each word
- ID: 54, Feature Details: Min character diversity of each word
- ID: 55, Feature Details: Max ratio of uppercase letters to all characters of each word
- ID: 56, Feature Details: Max ratio of digit characters to all characters of each word
- ID: 57, Feature Details: Max ratio of non-alphanumeric characters to all characters of each word

- ID: 58, Feature Details: Max length of the longest repeating character
- ID: 59, Feature Details: Max character length of words

- **Readability Feature Details**
  - ID: 1, Feature Details: Number of simple words features (with and without stopwords)
  - ID: 2, Feature Details: Number of complex words features (with and without stopwords)
  - ID: 3, Feature Details: Word length features (with and without stopwords)
  - ID: 4, Feature Details: Fog Index (FI) features (with and without stopwords)
  - ID: 5, Feature Details: Flesch Reading Ease Score (FRES) features (with and without stopwords)
  - ID: 6, Feature Details: SMOG index features (with and without stopwords)
  - ID: 7, Feature Details: FORCAST index features (with and without stopwords)
  - ID: 8, Feature Details: Flesch-Kincaid Readability Index (FKRI) features (with and without stopwords)
  - ID: 9, Feature Details: Simple Word FI features (with and without stopwords)
  - ID: 10, Feature Details: Inverse FI features (with and without stopwords)
  - ID: 11, Feature Details: SMOG-I feature
  - ID: 12, Feature Details: Automated Readability Index (ARI)
  - ID: 13, Feature Details: Coleman-Liau Index (CLI)

- **Lexical Diversity Feature Details**
  - ID: 1, Feature Details: Vocabulary Richness
  - ID: 2, Feature Details: Hapax legomena (V(1,N))
  - ID: 3, Feature Details: Hapax dislegomena (V(2,N))
  - ID: 4, Feature Details: Entropy measure
  - ID: 5, Feature Details: YuleK
  - ID: 6, Feature Details: SichelS
  - ID: 7, Feature Details: Honore

Vita

Candidate's full name: Dakota Staples
University attended:University of New Brunswick, 2018-2022, Bachelor of Computer Science

Publications:
W. Shahid, Y. Li, D. Staples, G. Amin, S. Hakak and A. Ghorbani, "Are You a Cyborg, Bot or Human?—A Survey on Detecting Fake News Spreaders," in IEEE Access, vol. 10, pp. 27069-27083, 2022, doi: 10.1109/ACCESS.2022.3157724.

Conference Presentations:
D. Staples, and S. Hakak, "Science Atlantic Mathematics, Statistics, and Computer Science Conference 2021" in Multilingual Phishing Email Detection Using Federated Learning

D. Staples, S. Hakak, and P. Cook, "20th Annual International Conference on Privacy, Security and Trust" in A Comparison of Machine Learning Algorithms for Multilingual Phishing Detection, 2023