Multilingual Phishing Email Detection Using Lightweight Federated Learning

1st Dakota Staples
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
dstaples@unb.ca

2nd Hung Cao
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
hcao3@unb.ca

3rd Saqib Hakak
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
saqib.hakak@unb.ca

4rd Paul Cook
Faculty of Computer Science
University of New Brunswick
Fredericton, Canada
paul.cook@unb.ca

***Abstract*—Given the escalating global threat of phishing emails, it is imperative to develop effective solutions to mitigate their potentially devastating impacts on society. This study endeavours to construct a federated multilingual spam detection system employing logistic regression, specifically targeting English, French, and Russian emails. This is the first work to the best of our knowledge which considers a non-deep learning setting for federated learning, and combines federated learning with multilingual phishing detection. Evaluation of the models is based on accuracy metrics which are compared with a most frequent class baseline. Our findings indicate that an optimal configuration comprises 10 clients undergoing 100 epochs of training with 100 rounds of federated learning, resulting in superior performance. Notably, this approach significantly outperforms the baseline, achieving an accuracy of 89.46% compared to 70%.**

***Index Terms*—phishing, machine learning, multilingual, federated learning**

I. Introduction

The widespread occurrence of phishing emails in modern society presents a notable concern, yielding numerous adverse outcomes when individuals are deceived by them. These deceptive emails afford hackers entry to accounts, encompassing email, social media, and online shopping platforms.

Previous research has already determined that phishing is not limited to English, and is a world-wide problem that spans many languages [1]. However, this problem has not slowed down or diminished. In 2023, there was a phishing campaign conducted in Japanese [2] as well as attacks in Hebrew and Arabic [3]. This highlights the continued importance of making sure that phishing email detection systems are capable of handling numerous languages.

Artificial intelligence (AI) detects abnormalities in phishing emails compared to normal benign emails, meaning that it can be utilized to determine if an email is malicious or not. In addition, AI, in particular machine learning (ML), has been shown to be effective at multilingual tasks [4] such as phishing detection [1]. Therefore, AI can be used to detect multilingual phishing emails.

Federated learning (FL) represents a decentralized approach to ML, wherein numerous edge nodes operate independently, undergoing training on local data samples instead of centralized ones as seen in conventional ML methods. This methodology serves to safeguard both data privacy and security [5]. FL operates by employing a global model alongside multiple edge devices. Initially, a global ML model undergoes training on a publicly accessible dataset. Following this, various edge devices download this same global model, enhancing their individual local models by leveraging input from each user. At regular intervals, each edge node transmits training updates to the global model, which then aggregates these updates, adjusting itself accordingly. Subsequently, the edge devices retrieve the updated global model, and the iterative process continues. Throughout this entire process, the global model remains oblivious to the specific data uploaded by users, thereby preserving their privacy.

Since ML can be used to detect emails in multiple languages, and FL can be used to preserve privacy, they can be combined to provide lightweight phishing detection to users who speak languages other than English. Previous research [6], [7], [8] has only focused on deep learning (DL), as opposed to lightweight models. Lightweight models, in the context of this research, refers to models that are not DL models.

To the best of our knowledge, this is the only research which not only uses a lightweight model in federated phishing detection, but examines multilingual phishing detection. To conduct the examinations, data in English, French, and Russian was obtained [9] and will be elaborated upon in subsequent sections of the paper. To perform the experiment, an FL system was built, and the hyperparameters were fine-tuned to achieve the best result. This work only considers lightweight models not only because of the research gap, but because deep learning models are more costly for the environment compared to lightweight models [10], [11].

This research aims to answer the following questions.

- **RQ1:** Which model is the best baseline for a multilingual FL system?
- **RQ2:** How does the number of rounds and epochs affect the multilingual FL system?
- **RQ3:** How does the number of FL clients affect the multilingual FL system?
- **RQ4:** Is FL viable for a multilingual phishing detection system?

This work was partially supported by the NSERC Discovery Grant (NSERC RGPIN 2025-00129

To address these research questions, our methodology involves a structured, four-step process for building and testing a multilingual federated phishing detection system. First, data is collected and preprocessed, primarily drawing from email datasets in English, French, and Russian. Then, we set up the federated learning (FL) environment using Flower FL [12]. For this, logistic regression is selected as a lightweight model. It also uses email features derived from EMFET [13] to facilitate feature extraction in multiple languages.

Our experiments address the key research questions by comparing eight models to establish a baseline, and by analyzing the optimal configuration for FL settings through varying clients and training rounds and epochs. Results are reported both centrally and in a distributed setting, ensuring comprehensive analysis across different configurations. This methodology not only explores multilingual phishing detection but also addresses FL challenges in a multi-language scenario which has not yet been done at the time of writing.

The rest of the paper is structured as follows: Section 2 delves into related and prior works. Section 3 outlines the methodology employed. Section 4 presents the results obtained. Section 5 offers concluding remarks on our work. Section 6 examines the limitations of this work and proposes some future directions.

II. Background

Multilingual federated phishing detection is an area yet to be explored leaving a gap in research. This section will highlight the existing work in the areas that have been done. There are 2 main areas which have been researched already: federated phishing detection and multilingual phishing detection. Research combining both the areas is needed which our research will begin to address.

A. Multilingual Phishing Detection

Vu et al. examine phishing emails across Vietnamese, Chinese, and English languages employing a rule-based strategy rooted in SpamAssassin, an Apache website, to classify and filter spam [14]. Good results were achieved with the utilization of 100 rules at a threshold of 0.5, resulting in a spam detection rate of 49.6% and a false alarm rate of 2.9% [15].

Researchers from Lithuania similarly explored the area of multilingual phishing email detection encompassing English, Russian, and Lithuanian languages. Their experiment tested naive Bayes, random forest (RF), and support vector machine (SVM). SVM emerged as the most effective, achieving an accuracy of 84%. However, the study suggests the potential of Deep Learning (DL) to yield even better results in subsequent research [16].

Although two papers are overviewed, additional work in the field of multilingual phishing detection has been done[17], [18], [19], [20].

B. Federated Learning

Thapa et al.'s research [7] investigates FL and phishing detection. They use three datasets (Enron spam [21], Nazario [22], and IWSPA-AP phishing emails) and compare two models (THEMIS and BERT). The findings suggest that while FL is feasible compared to centralized learning, it fails to match the performance of the latter. Additionally, the research highlights that an increase in the number of clients leads to a degradation in performance and a decrease in convergence speed for THEMIS, and vice versa for BERT.

Sun et al. propose a model named Federated Phish Bowl (FPB), integrating FL and long short-term memory (LSTM) [6]. For the dataset, emails from Microsoft 365 and the Enron dataset [21] are selected. FPB utilizes a five-layer model comprising three bidirectional LSTM layers, a fully connected layer with 200 neurons, and an output layer with one neuron. Results indicate that FPB surpasses individual client learning, with performance degradation observed as the number of clients increases.

In addition to the two studies examined above, there are also other pieces of work which examine federated learning and phishing detection [23], [24], [25].

C. Research Gaps

In general, the only two works to consider federated phishing detection did so with deep learning models, LSTM, THEMIS, and BERT. Deep learning models require increased resources to use compared to traditional models such as logistic regression in terms of computational power. They also have a higher environmental impact with regard to both carbon emissions and energy consumption [10], [11]. FL can be used for phishing detection due to its inherent privacy preserving nature. End users of a phishing detection system may have confidential emails that must be kept private, or they may simply want to increase their own privacy. To the best of our knowledge, there is no FL phishing detection work which uses these traditional models making this work even more novel.

III. Methodology

At a glance, our overall methodology is that we obtain our data, implement the server and client, and then conduct an experiment to choose the base model. Finally, we conduct experiments to test the number of clients and the number of rounds and epochs for training.

A. Data Used

Data acquisition in the form of multilingual phishing emails is required to perform the research. This work adopts the same data setup as previous work by [1].

The majority of the data comes from [9]. The other data will come from the Enron/Enron Spam and TREC 07 email datasets [[26],[21],[27]].

Three languages were chosen for the experiment, English, French, and Russian. For the English dataset, we utilized the Enron Spam dataset [26], a subset of the Enron dataset [21], specifically using the preprocessed variant of Enron1 [28]. For the French dataset, we sourced the spam segment from the prior research of Pan et al. [9]. However, this collection is comprised solely of spam emails and necessitated augmentation

with authentic ones. To achieve this, emails from the TREC07 dataset [27] were translated from English into French. This translation was performed using Google Translate via the API. Similarly, for the Russian dataset, the process closely mirrored that of the French dataset. Spam emails were acquired from Pan et al. [9], while legitimate emails (ham) were sourced from the preprocessed Enron Spam dataset [26] and subsequently translated from English into Russian.

The final training data file contained 3983 English emails (71% ham, 21% spam), 472 French emails (52% ham, 48% spam), and 175 Russian emails (50% for each) making 4630 emails total for training. The combined training set had a ham/spam ratio of 69% ham and 31% spam. For testing, 996 English, 472 French, and 175 Russian emails were used with the same splits as above for 1643 emails total. The combined test set had a ham/spam ratio of 70% ham and 30% spam. This means that our solution should surpass 70% accuracy as one could obtain 70% simply by choosing the most frequent class every time and thus will be our baseline.

While there is not much data compared to other datasets, especially in languages other than English, it works well as each client realistically would not have so much data at one time. This setting being researched envisions an area in which there are speakers of many languages, necessitating a multilingual phishing detection system to combat emails as emails could appear in different languages.

B. Implementation

For the implementation of FL, Flower FL will be used [12]. This is an FL framework that is model agnostic, meaning any model can be used. In this work, multiple clients will be ran on one machine. Future work could examine Flower's simulation module which can enable more clients.

FedAvg [29] is selected for our averaging strategy as this is a baseline for testing FL. Future work could try other popular algorithms and compare results, such as FedAvgM, FedProx, or FedBN.

The choice of model is needed as well. Where this is an edge environment, the devices may not have the resources to run a multilingual transformer-based model in practice such as XLM RoBERTa or GPT3. Hence, this work will examine how lightweight models work, such as logistic regression. In particular, an initial experiment will be performed to determine the base model for the FL system.

To start, the server sets the initial parameters for the base model (logistic regression), picks the strategy (FedAvg, and how many clients at minimum must be used for training), specifies how evaluation will occur, and starts the server along with setting how many rounds of training will happen which can be fine tuned. For evaluation, it reads the whole test set in for centralized evaluation, and after each round, updates the model accordingly and returns the loss and accuracy for that round. It also specifies a weighted average which is used to aggregate client results for distributed results.

First, the client takes in a partition id value to identify the client and it is used to obtain a partition of the data. The client then has to read the training and testing data in, and partition into X number of pieces (the number of clients total). Next, the model is created; logistic regression is chosen as experiments show that logistic regression performs the best in a non federated setting as shown in Figure 1. The L2 penalty is chosen, with X epochs which can be fine-tuned. The FL client is then created, defining how parameters are obtained, how training and model updating works, and how local evaluation occurs. The client is started and attempts to connect to the server. The utility file specifies helper functions, how to get model parameters, and how to set model parameters, and sets the parameters for the first time.

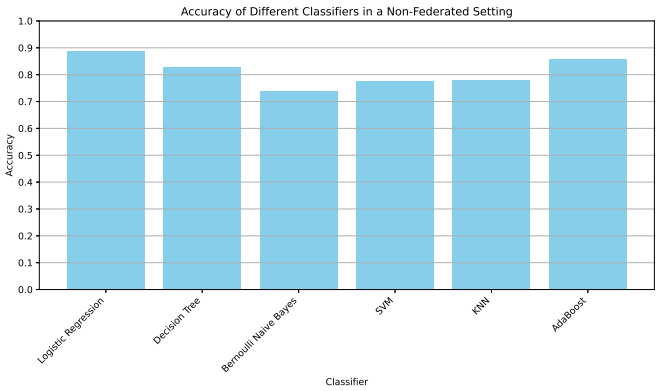

Fig. 1. Results showing the accuracy of the different lightweight models in a non-federated setting.

1) Lightweight-ness of Logistic Regression: Logistic regression is extremely lightweight compared to deep learning models. For this experiment, logistic regression has only 91 features, 90 plus a bias term. LLMs on the other hand, are multiple orders of magnitude larger even at the smaller level. A tiny LLM, named TinyBERT has "only" 14.5 million parameters [30]. The base BERT model has 110 million [31], and GPT-3 has 175 billion [32]. At the largest end of LLMs, they can reach parameter sizes of 1.2 trillion which is the size of GLaM [33]. From this, it is clearly observable that the logistic regression used in this research is extremely lightweight when compared to deep learning transformer based models.

C. Experimental Setup

The hardware components on which the experiments were ran are an Intel I7 4770K CPU, and 16 GB of RAM. Since the Scikit-Learn [34] library was used, no GPU can be used. However, since the models are lightweight, this is not an issue and no acceleration is needed.

In total, 3 experiments were ran: the initial experiment to find the base model, a 10-clients experiment to examine epochs and rounds, and an experiment with the ideal epochs and rounds but with varying clients. These experiments will address the research gap and provide us with the first research in the area of multilingual federated phishing detection.

The data for the below experiments is mixed English, French, and Russian phishing emails with features extracted using

EMFET [13]. The tool underwent research testing and exhibited strong performance with their data [35]. Our feature extraction process exclusively targets the body of the email, disregarding the header. A total of 59 features concentrates on various aspects of the email body itself, such as the overall tab count and the occurrence of designated spam words. The spam word list, integral to EMFET, has been translated into French and Russian to facilitate feature extraction in those languages. Additionally, 23 features are dedicated to assessing the readability of the body, encompassing metrics such as the tally of simple and complex words, along with the Simple Measure of Gobbledygook (SMOG) index. Furthermore, seven features are centered on lexical diversity, including the analysis of hapax legomena. These features are all listed in the original paper [13]. Although a total of 89 features are utilized, EMFET offers functionality to inspect both the header and attachments if necessary.

1) Experiment 1: The initial experiment examined eight total models: logistic regression, decision tree, random forest, Bernoulli Naive Bayes, SVM, KNN, adaboost, and an ensemble model with all of the classifiers. For random forest and SVM, the random state is set to the same each time. This is important as different states could give different results each time making comparisons difficult. Random forest also uses 100 estimators. KNN uses the floor of the square root of the total amount of emails. The accuracy is captured for all of the models, and compared against each other keeping the most frequent class baseline in mind. This experiment will address RQ1.

2) Experiment 2: This 10 client experiment was run 4 times, at 1 epoch and round, 10 epochs and rounds, 100 epochs and rounds, and 1000 epochs and rounds. Each client obtained a partition of the data and varying epochs and rounds. From this, the best combination was selected for further testing in the third experiment to see how clients affect results. The clients will be increased in intervals of 5, up to 60 where above, the system would start to give memory errors. Currently, 42 is used as the random state to shuffle the dataset; however, better random states may exist and worse random states may exist. In addition, each experiment is run 5 times and averaged to ensure the results are more robust. From this experiment, we can answer RQ3.

3) Experiment 3: For the third experiment, where the rounds and epochs are tested, the results are given in a distributed setting and a centralized setting. The centralized setting is where the FL server performs the testing with a central dataset and evaluates the global model on this dataset. The distributed setting is where the clients perform testing on themselves and send their results to the server for aggregation, in this case, a weighted average. The dataset is unique to each client. The centralized dataset is the testing dataset described above. The distributed dataset each client has is the same dataset but split into partitions. Finally, experiment 3 will address RQ2. RQ4 will be addressed by analyzing all of the experiments instead of one specific one.

IV. Results

In our research, we utilize accuracy as the primary evaluation metric. Nevertheless, we acknowledge the imbalance in class distribution. Although precision and recall could address this concern, we choose to interpret our findings relative to the baseline, represented by the most prevalent class.

A. Which Model Performs The Best in a Non-FL Setting?

Figure 1 shows the results for the non-federated setting. In this lightweight model setting, logistic regression performs the best, scoring 88.9%. This value will be used to compare against the federated setting's best value to see if there is an increase or decrease. Even in a non federated setting, this is a good value, surpassing the baseline value by 18.9%. Previous research on multilingual phishing detection has shown it is a difficult problem [1], so this result is a good start. From these results, logistic regression is chosen as the base model for the FL system and further testing is conducted on this.

B. How Do Epochs and Rounds Affect Performance?

First, the number of epochs and rounds was tested. In this setting, it was found that maximum performance comes from training logistic regression for 100 epochs and running the FL system for 100 rounds, achieving 89.46% accuracy in both the distributed setting and the centralized setting. Only the 1 round and 1 epoch setting had any difference between distributed and centralized. In general, the more rounds and epochs that are performed, the better the results. However, with this being said, 1000 epochs and rounds actually performs worse than 100. It can be assumed the model converges close to or around 100 and overfits when training for more.

C. How Does The Number of Clients Affect Performance?

From experiment 2, 100 rounds and epochs were found to have the best performance. Using this, and logistic regression, an experiment focusing on the number of clients was performed. It was found that 10 client setting performs the best, obtaining 89.44% accuracy on average. As a general trend, the number of clients is negatively correlated with accuracy. Average starts around 89% with a low number of clients, and with a high number of clients drops to around 86-87% accuracy. It is hypothesized that if the clients continued to grow, the difference would become more pronounced. At the current drop in accuracy, it is still viable to use with 60 clients.

D. Explainability of Features

Logistic regression being a lightweight non-transformer model means that compared to deep learning, it is much simpler to explain how the model came to a decision and which features are the most important. A logistic regression model is trained on the multilingual dataset, and then the coefficients for each feature are extracted. This process is repeated five times and averaged. This is performed in a non federated setting. The top and bottom three results will be shown in each table. Coefficient analysis is good for an initial evaluation which is what we provide here.

Table I show the results. From this, we can see to some extent which features are indicative of spam or ham emails. For spam, excessive punctuation (question marks) and odd structure (ratio of uppercase letters to lowercase letters) are a high indicator of spam. For ham emails, Sichel (a readability score) and good structure (entropy) are indicators of ham emails.

TABLE I
AVERAGE MULTILINGUAL FEATURE IMPORTANCE (COEFFICIENT)

Feature	Average Coefficient
MultipleQuestionMarks	2.08
RatioUpperLower	0.84
InverseFI_WithoutStopwords	0.63
RatioNonAlphaNumToAll	-0.63
Entropy	-1.12
Sichel	-1.66

E. Is FL Viable for a Multilingual Phishing Detection System?

Yes. FL is indeed viable for multilingual phishing detection according to the experiment shown in Figure 1, and described in experiments 2 and 3, meaning it is about on par with non FL, and surpasses the most frequent class baseline. However, there are more nuances to consider before this could be practically implemented. For the results in the above figures, the best accuracy is an average of 89%. This is a good accuracy, surpassing the most frequent class baseline of 70%. In addition, 89% is a very good accuracy considering the base model. Logistic regression is a very lightweight model, meaning a high accuracy when using it is very good. This means that when implemented, it is a lightweight model that nearly all edge devices would be able to support. In addition, logistic regression is much faster than deep learning models as they are much bigger and much more complex. While FL does appear to be viable, considering a zero shot learning setting like the one detailed in [1] would aide in proving its viability. Furthermore, in a practical system, privacy must be kept, in addition to considering other factors such as the hardware that the edge devices will be, and the latency between the devices and central server. There are also other factors such as when to update models, how often, and what devices to include.

V. CONCLUSION

This study explores the feasibility of utilizing federated learning in conjunction with a lightweight model for multilingual phishing detection. This is the first work to our knowledge which combines both multilingual phishing detection along with federated learning. Our findings indicate that FL, when combined with logistic regression as a base model, shows promising results in identifying phishing emails in English, French, and Russian. The achieved accuracy surpasses the most frequent class baseline, highlighting the potential of this approach to enhance email security for users across diverse linguistic backgrounds.

Our experiments highlight key considerations such as the impact of epochs, rounds, and the number of clients on system performance. We observed that optimizing these parameters is crucial for achieving the best results. Additionally, while FL demonstrates viability for multilingual phishing detection, practical implementation considerations such as privacy preservation, hardware constraints, and latency issues must be considered to ensure real-world effectiveness.

VI. LIMITATIONS

This work is limited by not sampling all clients at once and by how different random states affect the results. While the results are averaged, this is with the same random state each time, which may or may not have been a favourable state which is why testing more and averaging over the different states is future work which should be performed.

In addition, it is limited by not examining how more clients affect the results and if the accuracy continues to decrease even more. Flower FL [12] has a simulation module which could facilitate this for future work.

Another limitation is the assumption of data heterogeneity. In this research, all clients have the same amount of data samples. In a real world scenario, this is highly unlikely and unrealistic. Given this work considers multiple languages, the data each client receives could be very different in number and language.

Finally, a limitation is that part of the training data is obtained by translating English data, which is unideal. Emails in other languages could follow a different structure or wording which translation does not yield. However, while unideal, this was a deliberate and carefully considered decision to move the work forward and obtain results.

VII. FUTURE WORK

Future work could start by investigating further evaluation metrics such as precision, recall, or F1 which could provide an interesting further view on the findings.

Another direction could explore comparing against deep learning models. While the scope of this work is limited to lightweight models, other research may compare against deep learning to compare how much accuracy is lost against more powerful models.

One other future direction could be exploring overfitting further. While this work simply accepts overfitting occurs when going from 100 rounds and epochs to 1000, research could examine how to mitigate overfitting when training at higher values, including early stopping, regularization, or other federated learning algorithms.

Finally, an interesting direction would be to further explore the explainability of models. This could be performed by examining the odds ratio, as well as performing regularization or permutation importance testing.

REFERENCES

[1] D. Staples, S. Hakak, and P. Cook, "A comparison of machine learning algorithms for multilingual phishing detection," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, pp. 1–6, IEEE, 2023.

[2] M. Tobey, "Japanese train company shows need for multilingual nlu: Blog." https://www.safeguardcyber.com/blog/security/phishing-campaign-impersonating-japanese-train-company-shows-need-for-multilingual-nlu, Jan 2023.

[3] N. Grant, "Google builds on tech's latest craze with its own a.i. products." https://www.nytimes.com/2023/05/10/technology/google-ai-products.html, May 2023.

[4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 8440–8451, Association for Computational Linguistics, July 2020.

[5] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[6] Y. Sun, N. Chong, and H. Ochiai, "Federated phish bowl: Lstm-based decentralized phishing email detection," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 20–25, 2022.

[7] C. Thapa, J. W. Tang, A. Abuadbba, Y. Gao, S. Camtepe, S. Nepal, M. Almashor, and Y. Zheng, "Evaluation of federated learning in phishing email detection," *Sensors*, vol. 23, no. 9, 2023.

[8] C. Thapa, J. W. Tang, A. Abuadbba, Y. Gao, S. Camtepe, S. Nepal, M. Almashor, and Y. Zheng, "Evaluation of federated learning in phishing email detection." https://arxiv.org/abs/2007.13300, 2021.

[9] D. Pan, E. Poplavska, Y. Yu, S. Strauss, and S. Wilson, "A multilingual comparison of email scams," USENIX Association, Aug. 2020.

[10] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 3645–3650, Association for Computational Linguistics, July 2019.

[11] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the carbon emissions of machine learning." https://arxiv.org/abs/1910.09700, 2019.

[12] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework." https://arxiv.org/abs/2007.14390, 2020.

[13] Wadi Hijawi, H. Faris, Ja Far Alqatawna, I. Aljarah, Ala M. Al-Zoubi, and M. Habib, "Emfet: E-mail features extraction tool." http://rgdoi.net/10.13140/RG.2.2.32995.45603, 2017.

[14] Apache Software Foundation, "Apache SpamAssassin." https://spamassassin.apache.org/, 2025.

[15] L. Nguyen, D. Nguyen, L. Diep, V. Tuan, Q. A. Tran, and B. Lam, "Detecting vietnamese spams using a multi-objective evolutionary approach," *Research Gate*, 12 2017.

[16] J. Rastenis, S. Ramanauskaitė, I. Suzdalev, K. Tunaitytė, J. Janulevičius, and A. Čenys, "Multi-language spam/phishing classification by email body text: Toward automated security incident investigation," *Electronics*, vol. 10, no. 6, 2021.

[17] M. T. Banday and S. A. Sheikh, "Multilingual e-mail classification using bayesian filtering and language translation," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 696–701, 2014.

[18] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A new english/arabic parallel corpus for phishing emails," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 22, jul 2023.

[19] S. Salloum, *Enhancing Cybersecurity: Machine Learning and Natural Language Processing for Arabic Phishing Email Detection*. PhD thesis, University of Salford, UK 2023, 2024.

[20] J. Cao and C. Lai, "A bilingual multi-type spam detection model based on m-bert," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, 2020.

[21] L. Kaelbling, "Enron email dataset." https://www.cs.cmu.edu/ enron/, 2015.

[22] J. Nazario, "Index of / jose/phishing." https://monkey.org/ jose/phishing/, 2016.

[23] I. Ul Haq, P. Black, I. Gondal, J. Kamruzzaman, P. Watters, and A. Kayes, "Spam email categorization with nlp and using federated deep learning," in *International Conference on Advanced Data Mining and Applications*, pp. 15–27, Springer, 2022.

[24] B. Li, P. Wang, H. Huang, S. Ma, and Y. Jiang, "Flphish: Reputation-based phishing byzantine defense in ensemble federated learning," in *2021 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2021.

[25] S. Löbner, B. Gogov, and W. B. Tesfay, "Enhancing privacy in federated learning with local differential privacy for email classification," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology* (J. Garcia-Alfaro, G. Navarro-Arribas, and N. Dragoni, eds.), (Cham), pp. 3–18, Springer International Publishing, 2023.

[26] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," in *CEAS*, vol. 17, pp. 28–69, Mountain View, CA, 2006.

[27] G. V. Cormack and T. R. Lynam, "2007 trec public spam corpus." https://plg.uwaterloo.ca/ gvcormac/treccorpus07/, 2007.

[28] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Enron1." http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html, 2006.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (A. Singh and J. Zhu, eds.), vol. 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, PMLR, 20–22 Apr 2017.

[30] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *Findings of the Association for Computational Linguistics: EMNLP 2020* (T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 4163–4174, Association for Computational Linguistics, Nov. 2020.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[32] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, (Red Hook, NY, USA), Curran Associates Inc., 2020.

[33] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. P. Bosma, Z. Zhou, T. Wang, E. Wang, K. Webster, M. Pellat, K. Robinson, K. Meier-Hellstern, T. Duke, L. Dixon, K. Zhang, Q. Le, Y. Wu, Z. Chen, and C. Cui, "GLaM: Efficient scaling of language models with mixture-of-experts," in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 5547–5569, PMLR, 17–23 Jul 2022.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[35] W. Hijawi, H. Faris, J. Alqatawna, A. M. Al-Zoubi, and I. Aljarah, "Improving email spam detection using content based feature engineering approach," in *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pp. 1–6, 2017.