Enhancing Phishing Email Detection with Context-Augmented Open Large Language Models

Fabian Nicklas[1], Nicolas Ventulett[2], and Prof. Dr.-Ing. Jan Conrad[3]

[1] University of Applied Sciences Kaiserslautern
fani1001@stud.hs-kl.de
[2] University of Applied Sciences Kaiserslautern
nive1002@stud.hs-kl.de
[3] University of Applied Sciences Kaiserslautern
jan.conrad@hs-kl.de

Abstract. Large Language Models offer a promising approach for improving phishing detection through advanced natural language processing. This paper evaluates the effectiveness of context-augmented open LLMs in identifying phishing emails. An approach was developed that combines the methods of Few-Shot Learning and Retrieval-Augmented Generation (RAG) to remarkably improve the performance of LLMs in this area. On this basis, it has been shown that the presented approach can significantly improve the recognition rate even for smaller models.

Keywords: Artificial Intelligence, AI, Cybersecurity, Large Language Models

1 Introduction

Phishing is a significant and increasing threat to cybersecurity. Attacks using constantly evolving techniques aim to tempt people into revealing sensitive personal information. It is estimated that 90 percent of all successful cyberattacks have phishing as an initial vector of attack [1]. The rise of Large Language Models (LLM) has revolutionized the field of Natural Language Processing (NLP). First popular representatives as the model GPT (Generative Pretrained Transformer) by OpenAI [2] have showcased the power of Large Language Models for language generation and understanding. They are trained across diverse datasets of large text corpora and their application beyond the original task of text generation for machine learning problems is an increasingly addressed research question [3]. LLMs with their deep understanding of natural language are a promising starting point for the detection of phishing emails. This paper presents an approach of combining the in-context learning and augmentation methods Few-Shot Learning and Retrieval Augmented Generation (RAG) for phishing email classification. It dynamically augments the context of LLMs in a problem-specific way at the time of inference without the need for intensive, task-specific training or the use of a dedicated model. The approach is evaluated in experiments across different open models and compared to more common state-of-the-art prompting techniques.

2 Related Work

Phishing email detection has been an area of active research for decades, evolving from the application of rule-based systems, lexical analysis and machine learning algorithms, including SVM and tree-based classifiers, up to leveraging deep learning methods like recurrent and convolutional neural networks as well as transformers [4][5]. The use of Large Language Models for identifying email phishing is still an emerging field with a sparse number of research publications.

A majority of recent studies based their work on the GPT models of OpenAI [6][7][8]. The model family achieved a high level of popularity with the release of its derivative ChatGPT. Rosa et al. [7] achieved an overall accuracy of 75.75 % for binary phishing email classification by feeding emails to GPT 3.5. With their high number of active parameters the GPT models proved a strong performance across many application areas, however, the GPT models are proprietary and closed-source [9]. This paper focuses on the use of open models, that are free to use and are meeting higher demands regarding data privacy. While some studies on phishing detection use open LLMs solely as upstream feature extractors for other machine learning methods [3], Koide et al. [10] employs the model Llama 2 to classify emails and achieves an overall accuracy of 88.61 % through prompt engineering. Their study contrast this with the use of the much-larger GPT-4 model showing 99.70% accuracy.

Baumann et al. [11] proposes a combination of RAG and FSL to generate models for domain-specific languages (DSLs) finding application in the field of software engineering. Their approach uses RAG to retrieve relevant examples from a knowledge base, enabling FSL to generate synthetic models for underrepresented DSLs lacking sufficient training data and thereby adapting a LLMs output syntax. Literature review showed, the method of using a fusion of RAG and FSL to improve a LLM's capability to solve unknown machine learning tasks has not been addressed to date.

3 Methodology

3.1 Dataset

The experiments conducted in this study aim to evaluate the performance of the proposed approaches for the classification of phishing emails. For this purpose, a dataset containing both phishing and legitimate emails was created by concatenating two publicly available datasets. The *CSDMC Spam Corpus*[12] includes 2,949 so-called "ham emails", legitimate messages that do not fall into the categories of phishing or "spam". It has already been used in similar studies as [10]. The phishing emails were sampled from the *Phishing Pot* [13] dataset and are real emails collected from August 2022 to July 2024. In contrast to [14] this approach do not include synthetic phishing samples or emails collected well into the past, as in [15]. By choosing an up-to-date source dataset newer phishing techniques are also represented in our final dataset. From each source dataset 2,900 emails were randomly sampled to build a new set with a total of 5,800 emails, balanced between the two classes *phishing* and *no phishing*. Samples with an email body of less than 50 characters or more than 420,000 characters were not considered as valid samples and were discarded in the selection step. In a subsequent data-cleaning process, all non-ASCII characters in the messages were removed. Each message sample consists of the concatenation of the email's subject and its body. If the message body was available in text format and the HTML format, this approach prioritized the HTML part and converted to plain text by removing all HTML-related fragments. This study does not address the role of email attachments as an attack vector, all attachments included in the samples were removed.

3.2 Model Selection

The experiments were evaluated for a variety of Large Language Models that represent the current state of the art and are published under an open license. The approach deliberately refrained from the use of commercial models such as GPT4 (OpenAI). The selected AI models are OpenChat 7B [16], Mixtral 8x7B [17], Mistral 7B [18], Gemma2 9B and 27B (Google Deep Mind) [19], Llama3.2 1B and 3B [20], Mistral-Small 22B [18], Command-R 35B [21], as well as Llama3.1 8B and Llama3.1 70B (Meta AI) [20]. All models were pre-trained by their respective authors on different datasets and differ in their architecture and the number of parameters. While models with a larger number of parameters generally have a greater ability to understand complex patterns and relationships, they may tend to show over-fitting behavior and be less applicable to new and unseen data.

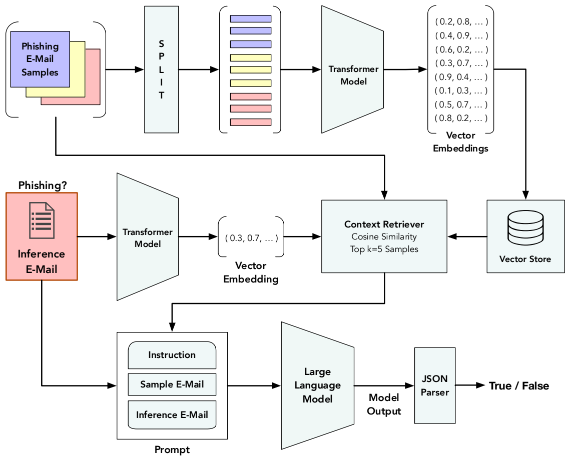

Fig. 1. Fusion of Few-Shot Learning (FSL) and Retrieval-Augmented Generation (RAG).

3.3 Detection of Phishing Emails with Large Language Models

Following the creation of the dataset, two different prompts[4] problem-specific prompts were developed to guide the LLMs to perform the classification task and produce the desired output. The same prompts were used across all AI models to allow for a consistent evaluation process. A fundamental prompt is provided for the evaluation of an email (**Prompt1**):

```
You are an expert in detecting phishing emails.
Your task is to determine whether it is a phishing email or not.
You are not supposed to justify or explain your decision.
{format_instructions}
E-mail:
‘‘‘{email}‘‘‘
```

[4] https://github.com/n-vent/llm-phishing-detection-paper

In the prompt, the LLM is given a fictitious role and the specific task to perform. The structure follows the *Persona Pattern*, a commonly used instruction scheme [22] that is independent of the area of application and the choice of model. **Prompt 2** extends the first prompt with a list of characteristics that may proof a phishing attempt. These include characteristics such as an impersonal generic greeting, urgent calls for action or demanding personal information such as the victim's bank details.

The evaluated models are trained to generate textual output in natural language. The models are guided to generate structured output by providing additional formatting instructions as JSON schema [23] in the prompt. At the time of inference, the passed sequence of prompt, preprocessed email and formatting instructions is concatenated and used as model input. The email is inputted directly into the language model, without the need for any feature extraction. A subsequent parser extracts the result of the classification from the model's text output as JSON, holding a boolean variable. The simple architecture of this approach does not include any components other than the described input construction, the respective language model and the JSON parser.

3.4 Context Augmentation through Few-Shot Learning and Retrieval-Augmented Generation for Improved Phishing Detection

This paper presents an improved approach for the classification of phishing emails by augmenting the knowledge of an already trained Large Language Model in-context and at the time of inference. The approach combines the methods of *Few-Shot Learning* (FSL) [24] and the *Retrieval Augmented Generation* (RAG) [25]. With FSL, also referred to as in-context learning, the model receives task demonstrations in natural language for a considered problem as part of the passed prompt. This allows the pre-trained models to address unknown tasks without a comprehensive training process or fine-tuning. FSL has shown to extend a language model's capability outside of the data is has been trained on [11].

Instead of prompting prepared and static FSL examples of phishing emails, it is proposed to dynamically select a relevant set of examples at the time of inference based on the input email. Using the technique of Retrieval Augmented Generation, examples from a knowledge base are selected and integrated into the prompt before generation. The LLMs gain access to domain-specific information that was not present in their training data. The model does not persistently store the augmented information and its parameters remain unchanged.

Figure 1 shows the architecture of the proposed RAG FSL fusion approach. A collection of examples of phishing emails serve as the RAG knowledge source and are individually split into blocks with a maximum length of 200 characters. By leveraging a transformer model, vector embeddings are obtained for each block as numerical vectors that represent semantic relationships. The pre-trained embedding model *Sentence-BERT* (all-MiniLM-L6-v2) was selected as the transformer, which maps natural language sentences into a 384-dimensional vectorspace and is commonly used for semantic search [26]. A vector database is populated with the generated embeddings for the 200-character-blocks. For the classification of a suspicious email, it is encoded into its vector representation using the same transformer model in order to find similar phishing examples to use in FSL. A set of relevant examples is selected by the maximum possible semantic similarity via a vector search. The cosine similarity [27] between the email embedding and each vector of blocks in the database is considered as a similarity measure (context retriever). The k most similar samples from the phishing dataset with the lowest semantic distance are selected as relevant FSL examples.

The prompt provided to the model is a concatenation of the instruction for the prediction task and the output schema, $k = 5$ positive phishing examples for in-context augmentation and the email message to be classified (**RAG FSL**):

```
You are an expert for detection of phishing emails.
For example, the following emails are phishing emails:
    Example 1 is a phishing email: {sample-email-1}
    [...]
    Example k is a phishing email: {sample-email-k}
Your task is to scan the following email, to decide whether or not
it is a phishing email and to use the provided JSON schema
for answering the question.
{format-instructions}
Question: Is the following email a phishing mail?
‘‘‘{email}‘‘‘
```

The result is extracted from the model output in a structured form using a JSON parser in the same way as the first approach (see section 3.3).

4 Experiments and Results

The 5,800 emails in the constructed dataset were systematically shown to the language models and the result of the classification evaluated for each sample. Each email was processed with **Prompt 1**, the extended **Prompt 2** as well as the proposed approach **RAG FSL**. Each prompt was evaluated across all of the selected eleven models. The model temperature parameter was set to 0.0, determining whether the output is more creative and random or more predictable. Other hyper-parameters than temperature were not changed. A total of 191,400 classification were run in this study. Model inference was performed on a NVIDIA A100 GPU with 80 GB of memory. In the evaluation, the performance of the approaches is assessed and various models using the quantitative metrics of precision, recall, F1 score, specificity and classification accuracy. If no valid JSON-data could be parsed from the model output by, the result was discarded in the evaluation. This could lead to an unbalanced number of positive and negative classes, which is met by calculating the metrics weighted by the number of samples as defined in [28]. The 2,900 emails of the positive class in the phishing dataset serve as knowledge source for RAG FSL. To guarantee the validity of the evaluation results and prevent target leakage, it was verified that the RAG phishing sample were not equal to the email test candidate at prediction time.

Table 1 shows the results of the conducted experiment to evaluate the presented approaches for the phishing email classification problem. The variance in results of the individual LLMs across the three different experiments shows the influence of the different model architectures and their training datasets on the performance in this classification task. The size of the models, measured by the number of active parameters, can be seen as an important but not decisive factor in the accuracy of the prediction. While Llama3.1 70B with 70 billion parameters always clearly outperforms the smaller 8B variant, the model Mixtral 8x7B with 13 billion active parameters already achieves an accuracy of 94.71% with prompt 1 compared to Llama3 with 92.82%.

The LLMs with less than 10 billion active parameters show a mostly higher performance using prompt 1 or prompt 2. When using RAG FSL on smaller models, the quality of the prediction seems to decline. Gemma2 9B can be noted as an exception, since the

Table 1. Performance of different Large Language Models for phishing email classification for *Prompt 1*, the extended *Prompt 2* and the proposed context-augmenting *RAG FSL* fusion approach. Models are in ascending order by their number of active parameters.

Model	Experiment	Precision	Recall	F1	Specificity	Accuracy
Llama3.2 1B	Prompt 1	41,94 %	50,39 %	33,86 %	**99,86** %	50,39 %
	Prompt 2	51,09 %	**51,08** %	**51,07** %	50,27 %	**51,08 %**
	RAG FSL	**54,91** %	**51,08** %	38,28 %	96,33 %	**51,08 %**
Llama3.2 3B	Prompt 1	68,32 %	**53,77** %	**41,76** %	98,58 %	**53,77 %**
	Prompt 2	**74,02** %	51,40 %	35,71 %	**99,96** %	51,40 %
	RAG FSL	63,92 %	52,09 %	37,47 %	99,14 %	52,09 %
OpenChat 7B	Prompt 1	91,19 %	**91,16** %	**91,16** %	89,92 %	**91,16 %**
	Prompt 2	87,53 %	84,08 %	83,72 %	69,10 %	84,08 %
	RAG FSL	**91,43** %	90,43 %	90,37 %	**98,15** %	90,43 %
Mistral 7B	Prompt 1	87,43 %	85,99 %	85,84 %	95,87 %	85,99 %
	Prompt 2	89,52 %	**89,38** %	**89,37** %	92,39 %	**89,38 %**
	RAG FSL	**89,91** %	88,13 %	88,01 %	**98,65** %	88,13 %
Llama3.1 8B	Prompt 1	**87,82** %	**87,70** %	**87,69** %	90,68 %	**87,70 %**
	Prompt 2	83,88 %	77,76 %	76,57 %	99,18 %	77,76 %
	RAG FSL	78,91 %	66,24 %	61,85 %	**99,44** %	66,24 %
Gemma2 9B	Prompt 1	93,43 %	92,86 %	92,84 %	87,27 %	92,86 %
	Prompt 2	94,44 %	94,30 %	94,29 %	91,54 %	94,30 %
	RAG FSL	**95,16** %	**95,00** %	**95,00** %	**92,01** %	**95,00 %**
Mistral-small 22B	Prompt 1	94,97 %	94,54 %	94,53 %	99,43 %	94,54 %
	Prompt 2	93,64 %	92,85 %	92,81 %	**99,61** %	92,85 %
	RAG FSL	**95,79** %	**95,66** %	**95,66** %	98,23 %	**95,66 %**
Gemma2 27B	Prompt 1	95,55 %	95,49 %	95,48 %	93,62 %	95,49 %
	Prompt 2	95,97 %	95,97 %	95,97 %	95,36 %	95,97 %
	RAG FSL	**96,15** %	**96,12** %	**96,12** %	**97,28** %	**96,12 %**
Command R 35B	Prompt 1	90,88 %	90,36 %	90,33 %	96,06 %	90,36 %
	Prompt 2	91,84 %	91,71 %	91,71 %	94,50 %	91,71 %
	RAG FSL	**93,91** %	**93,45** %	**93,43** %	**98,58** %	**93,45 %**
Mixtral 8x7B	Prompt 1	**94,85** %	**94,71** %	**94,70** %	97,54 %	**94,71 %**
	Prompt 2	92,26 %	92,02 %	92,01 %	88,32 %	92,02 %
	RAG FSL	94,33 %	93,88 %	93,87 %	**98,88** %	93,88 %
Llama3.1 70B	Prompt 1	93,54 %	92,82 %	92,79 %	99,26 %	92,82 %
	Prompt 2	92,42 %	91,27 %	91,21 %	**99,54** %	91,27 %
	RAG FSL	**96,22** %	**96,18** %	**96,18** %	97,69 %	**96,18 %**

RAG FSL approach provides the best results. According to the data produced during the experiments Llama3.2 1B as well as 3B show behavior of randomly guessing and an overall lack of capability for the given task. Event tough the model size is small in parameters, the RAG FSL combination leads to an accuracy of 95% using Gemma2 9B which performs remarkably well in comparison to the results of larger models.

Prompt 2 is not capable of improving the performance of the models noticeably. On the contrary, it produces more inaccurate classification results than the more fundamental prompt 1. Only Mistral 7B shows improvements using prompt 2, outperforming other approaches.

The larger models consistently perform better using RAG FSL. The only exception is Mistral 8x7B which achieves its best classification results using prompt 1. An remarkable conspicuity is that the larger models while using the RAG FSL only marginally outperform Gemma2 9B by an accuracy delta of 1.18%. This leads to the conclusion that after a certain point model size doesn't improve the results very much, but leads to increased resource consumption. Some of the larger models even perform worse than Gemma2 9B when looking at the F1-score, the ones performing better do this only by a very small percentage.

The proposed approach for context reinforcement using a fusion of FSL and RAG outperforms the results of prompt 2 in the experiments for most of the larger models with the exception of Mixtral 8x7B which performs best using prompt 1.

A maximum accuracy of 96.18% is achieved with the Llama3.1 70B model. The performance of Llama3.1 70B increased the most, from 92.82% and 91.27% to 96.18% accuracy, with a reduced false negative rate. Also the performance of the much smaller Gemma2 9B improves from 92.86% and 94.30% to 95.00% accuracy.

The results show that choosing the right model and methodology is crucial for the effectiveness of phishing detection. It can be concluded, that most smaller models lack the capability of using the RAG effectively in the context of phishing detection.

5 Conclusion and Future Work

This work evaluates how well LLMs are able to distinguish legitimate emails from phishing emails. The paper presents an approach that improves the effectiveness of detection by combining the methods of Few-Shot Learning and RAG for contextual reinforcement. The knowledge of the language model is dynamically enhanced at the time of inference by in-context and problem-specific learning without the need of computationally intensive adjustments to the actual AI model and its parameters. Experiments on a generated test dataset have shown that our approach significantly increases the recognition rate of models with fewer parameters and lower resource requirements, and outperforms previous approaches using open LLMs. This approach achieves an accuracy of 96.18% for the classification of phishing emails.

The results of this work raise further questions for future research on the detection of phishing emails with LLMs. In a next step, it should be investigated how a fusion of the RAG information source with additional data sets affects the detection accuracy. A promising approach could be the generation of phishing examples by an LLM itself, as already used by attackers. In addition, the use of other embedding models and different semantic search methods should be evaluated. It would also be useful to consider email metadata and file attachments. Furthermore, agent approaches that extend the capabilities of LLMs with functional tools, e.g. for retrieving API interfaces, could be investigated.

References

1. Cloudflare: Bericht zu Phishing-Bedrohungen 2023. Technical report, München (2024)
2. OpenAI, Achiam, J., et al., S.A.: Gpt-4 technical report (2024)
3. Nahmias, D., Engelberg, G., et al., D.K.: Prompted contextual vectors for spear-phishing detection. (2024)
4. Crawford, M., Khoshgoftaar, T., Prusa, J.e.a.: Survey of review spam detection using machine learning techniques. Journal of Big Data **2** (10 2015) 23
5. Thakur, K., Ali, M.L., Obaidat, M.A.e.a.: A systematic review on deep-learning-based phishing email detection. Electronics **12**(21) (2023)
6. Roumeliotis, K.I., Tselikas, N.D., Nasiopoulos, D.K.: Next-generation spam filtering: Comparative fine-tuning of llms, nlps, and cnn models for email spam classification. Electronics **13**(11) (2024)
7. Rosa, S., Gringoli, F., Bellicini, G.: Hey chatgpt, is this message phishing? (06 2024) 1–10
8. Heiding, F., Schneier, B., et al., A.V.: Devising and detecting phishing: Large language models vs. smaller human models (2023)
9. Hou, X., Zhao, Y., et al, Y.L.: Large language models for software engineering: A systematic literature review (2024)
10. Koide, T., Fukushi, N., et al., H.N.: Chatspamdetector: Leveraging large language models for effective phishing email detection. (2024)
11. Baumann, N., Diaz, J.S., Michael, J.e.a.: Combining retrieval-augmented generation and few-shot learning for model synthesis of uncommon dsls. Modellierung 2024 Satellite Events (2024)
12. Zhang, R.: Csdmc2010 spam corpus. International Conference on Neural Information Processing. https://github.com/zrz1996/Spam-Email-Classifier-DataSet (2010)
13. Anonymous: Phishing pot dataset. https://github.com/rf-peixoto/phishing_pot. (2024)
14. Jamal, S., Wimmer, H., Sarker, I.H.: An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. SECURITY AND PRIVACY **n/a**(n/a) (2024) e402
15. Patel, H., Rehman, U., Iqbal, F.: Large language models spot phishing emails with surprising accuracy: A comparative analysis of performance. (2024)
16. Wang, G., Cheng, S., Zhan, X.e.a.: Openchat: Advancing open-source language models with mixed-quality data. arXiv preprint arXiv:2309.11235 (2023)
17. Jiang, A.Q., Sablayrolles, A., et al., A.R.: Mixtral of experts. (2024)
18. Jiang, A., Sablayrolles, A., et al., A.M.: Mistral 7b. (2023)
19. Gemma Team, G.D.: Gemma 2: Improving open language models at a practical size. (2024)
20. Touvron, H., Lavril, T., et al., G.I.: Llama: Open and efficient foundation language models. (2023)
21. Gomez, A.: Command r: Retrieval-augmented generation at production scale. (2024)
22. White, J., Fu, Q., Hays, S.e.a.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv e-prints (February 2023) arXiv:2302.11382
23. Pezoa, F., Reutter, J.L., Suarez, F.e.a.: Foundations of json schema. In: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee (2016) 263–273
24. Brown, T.B., Mann, B., Ryder, N.e.a.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20 (2020)
25. Lewis, P., Perez, E., Piktus, A.e.a.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20 (2020)
26. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. (2019)
27. Singhal, A.: Modern information retrieval: A brief overview. IEEE Data Eng. Bull. **24** (2001) 35–43
28. Pedregosa, F., Varoquaux, G., Gramfort, A.e.a.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12** (2011) 2825–2830