APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users

Giuseppe Desolda[1], Francesco Greco[1,*] and Luca Viganò[2]

[1] University of Bari "A. Moro", Via Orabona 4, 70125 Bari BA, Italy
[2] Department of Informatics, King's College London, Bush House, 30 Aldwych, London WC2B 4BG, UK

Abstract

Phishing is one of the most prolific cybercriminal activities, with increasingly sophisticated attacks. It is, therefore, imperative to explore novel technologies to improve user protection across both technical and human dimensions. Large Language Models (LLMs) offer significant promise for text processing in various domains, but their use for defense against phishing attacks still remains scarcely explored. This paper presents APOLLO, a tool based on OpenAI's GPT-4o to detect phishing emails and generate warnings to protect users. These warnings contain tailored explanations about why a specific email is dangerous, which can improve users' decision-making capabilities. We evaluated APOLLO's performance in classifying phishing emails and found that GPT-4o obtains 97.40% accuracy in the task; this performance can be further improved by integrating data from third-party services, resulting in a near-perfect classification rate (99.99% accuracy). To assess the perception of the explanations generated by this tool, we also conducted a study with 20 participants, comparing four different explanations presented as phishing warnings. We compared the LLM-generated explanations to four baselines: a manually crafted warning, and warnings from Chrome, Firefox, and Edge browsers. The results show that not only the LLM-generated explanations were perceived as high quality but also that they can be more understandable, interesting, and trustworthy than the baselines. These findings suggest that using LLMs as a defense against phishing is a very promising approach and APOLLO represents a proof of concept in this research direction.

Keywords

Phishing, LLMs, Warnings, Explanations, Email classification

1. Introduction

In a world that is becoming increasingly dependent on its digital counterpart, phishing attacks pose a substantial risk to users, organizations, and IT systems. Phishing is indeed one of the most used attack vectors for various purposes, like stealing credentials and spreading malware [26]. These attacks are mainly effective because they leverage human vulnerabilities that can be potentially found in every user, such as their lack of knowledge, stress, and lack of time [17]. Therefore, as phishing attacks continue to get more complex and sophisticated, there is an urgent need to enhance the effectiveness of phishing protections, addressing both technological cybersecurity solutions [2, 28] and human aspects of security [17, 34].

Technical defenses often include the use of machine learning (ML) models to detect phishing content and can be used to classify malicious emails or websites with high accuracy [2, 28]. On the human side, warning dialogs are often used to alert users about potential threats when their systems (e.g., browsers or email clients) detect suspicious content. However, warnings are often ineffective for several reasons, such as not being relevant to users [1]; by including explanations in warnings, users can be aided in deciding whether to trust suspicious content, gain trust in the system and be motivated to heed the warning [7, 9, 44]. Nonetheless, generating meaningful and user-friendly warnings with explanation messages is a cumbersome task and requires significant human effort [16], is error-prone and not easily scalable over time to adapt to new phishing attacks. Moreover, to

* Corresponding author.
giuseppe.desolda@uniba.it (G. Desolda); francesco.greco@uniba.it (F. Greco); luca.vigano@kcl.ac.uk (L. Viganò)
0000-0001-9894-2116 (G. Desolda); 0000-0003-2730-7697 (F. Greco); 0000-0001-9916-271X (L. Viganò)

generate explanations with traditional ML models, these need to be interpretable and require datasets for training [22], which might not always be easily available.

An intriguing avenue in cybersecurity is leveraging Large Language Models (LLMs) as a proactive defense tool against phishing attacks. LLMs such as OpenAI's GPT have showcased remarkable abilities in detecting patterns and anomalies in emails or websites indicative of potential phishing attempts with surprising accuracy [23, 29]. Moreover, the prospect of using LLMs to generate warning messages and explain the model's suspicions could significantly enhance protection against phishing threats. This approach, being considerably faster than the manual design of explanations [16], would dynamically produce personalized explanations that vary based on the phishing content.

This paper presents the findings of recent research conducted to investigate the potential of LLMs in detecting phishing emails and to address the limitations of existing warnings. In particular, we illustrate APOLLO (*Advanced Phishing preventiOn with Large Language model-based Oracle*), a tool based on OpenAI's GPT-4o [38] and other third-party services for the automatic classification of phishing emails and for the generation of explanation messages that describe the reasons why an email is suspicious and should not be trusted. The tool's name was derived from the Greek deity Apollo, the god of prophecy, which recalls its capability to make predictions about emails alerting users in case of phishing attempts. To assess the tool's capabilities in detecting phishing emails, we performed a thorough evaluation process that also considered challenging scenarios in order to measure the performance of GPT-4o as is, augmented with external information [31], and primed toward erroneous classifications.

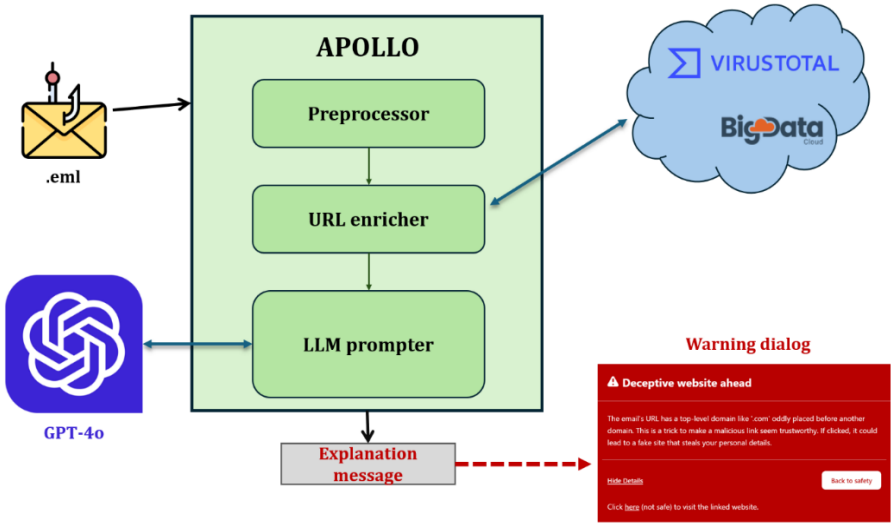

Figure 1. The architecture and information flow of APOLLO

We proceed as follows. Section 2 discusses related work on warning dialogs and the use of LLMs as a defensive tool for phishing. In Section 3, we describe the architecture and functioning of the APOLLO tool. Section 4 discusses the evaluation process of APOLLO using GPT-4o to assess the tool's performance in classifying phishing emails. Section 5 reports the user study conducted with 20 users to validate the warnings generated by APOLLO. Section 6 discusses the results of the study. Section 7 concludes the paper and presents possible future work.

2. Related Work

2.1. Phishing Defense

Different countermeasures can be employed to protect users from phishing attacks. A popular strategy is based on technology barriers designed to automatically detect and eliminate phishing emails and websites [19, 25]. ML solutions are often used to classify content automatically as genuine or phishing. Blocklists (or deny lists) are another phishing detection technique used to filter out

malicious websites that are found on these lists. These can be useful to detect malicious websites with high precision since blocklisted sites are almost definitely malicious, especially if the list is manually updated; however, blocklists are not effective at detecting zero-day attacks, as they require time to be updated. Contrarily, ML solutions allow the detection of even brand-new phishing content, as an AI model learns what are the critical indicators of malicious emails and websites to filter them out.

Eliminating the phishing threat is currently impracticable with automated methods, as no existing tools detect phishing websites or emails with 100% accuracy [18, 21]. Moreover, trying to minimize the number of false negatives (i.e., undetected phishing emails) would increase the number of false positives (i.e., genuine emails classified as phishing ones); this would risk jeopardizing users' productivity, as they would eventually lose genuine emails due to misclassification [40]. Therefore, to avoid blocking contents that could potentially be relevant for the user but that are suspicious, they can be instead shown together with a warning dialog that alerts users of possible dangers, leaving the final decision up to them [30]. Warning dialogs, however, are often ineffective for a variety of reasons [1]. For example, poor warning effectiveness is also due to a lack of explanations about the specific risk related to the phishing attack [7]. Including explanations in warnings has been shown to help users (who are often non-experts in cybersecurity) understand the danger and make more informed decisions [17].

2.2. LLMs for Phishing

Large Language Models (LLMs) are considered one of the most significant technological advancements in recent years. LLMs can achieve human-like performance in various human tasks [25] also thanks to their large number of parameters, which enable them to identify intricate patterns in linguistic data. There are several commercial LLMs, the most popular being OpenAI's GPT models, including GPT-4o and o1, Google's Gemini 2.0, Anthropic's Claude 3.5 Sonnet, and Meta's Llama 3.3.

Interaction with LLMs usually happens in natural language using messages called "prompts". *Prompt engineering* guidelines and techniques allow to optimize the performance of LLMs by writing proper prompts [15, 41]; an example of such a technique is the "few-shot prompting" [13], i.e., providing the LLM with examples of input-output interactions to define the structure and style of the desired output.

LLMs can be exceptionally good at performing tasks in a "zero-shot" way, i.e., without any specific training data [36]. This peculiarity might address the possible lack of datasets needed to train traditional ML models. Recently, LLMs have been used as cybersecurity tools: an example of such an application is the analysis of malicious scripts to explain the potential danger to cybersecurity analysts [20]. Another security application is in the context of phishing, where LLMs can be used to process and classify textual information like emails and websites to detect phishing content. GPT-2 has been shown to be promising in the task of detecting phishing emails in earlier studies [35], including adversarial settings [21]. The technological advance in LLMs seems to improve the phishing detection task substantially. For example, [29] Koide et al. proposed a method to detect phishing websites with LLMs, comparing GPT-3.5 and GPT-4; the latter model achieved substantially better results compared to GPT-3.5, achieving a 98.4% accuracy (vs. the 92.6% accuracy of GPT-3.5), with a significant improvement in GPT-4 in avoiding false negatives. In the work of Heiding et al. [23], four LLMs (GPT-4, Claude-1, Bard, and LLaMA2) were used to detect the intention (malicious vs. genuine) of phishing emails generated by humans and by GPT-4. All the LLMs proved to be able to detect malicious content, even in non-obvious phishing emails, effectively.

3. APOLLO: an LLM-powered System for Defense Against Phishing Attacks

APOLLO is a tool written in Python 3.10.12 and powered by GPT-4o to (i) classify an email as phishing or legitimate and (ii) generate an explanation for the user in case of a phishing email. As

an LLM model, we chose to use GPT-4o because, at the time of writing this article, it resulted in the best-performing model for classification-like tasks[2], such as the one performed by our system. The source code of APOLLO can be found at this link. APOLLO comprises 3 main modules (see), which are detailed in the next subsections: the *preprocessor* module, the *URL enricher* module, and the *LLM prompter* module.

Figure 2 shows an example of a warning generated by APOLLO explaining that an email was deemed to be malicious because it contains a URL with a top-level domain ".com" misplaced (the phishing URL was "https://amazonservices.com.cz/account.php"). The complete list of emails and warnings can be found in the APOLLO repository at this link.

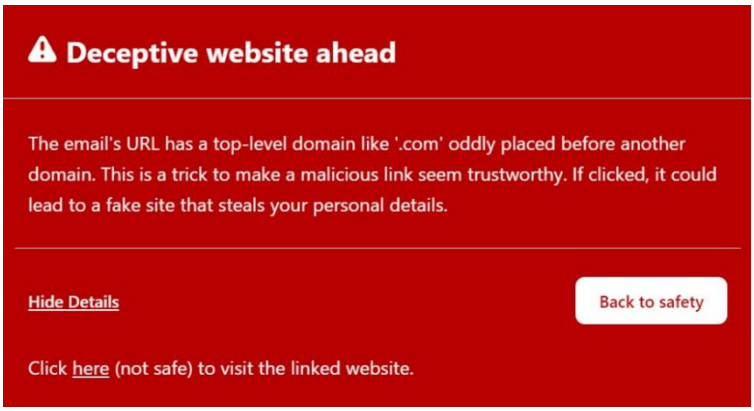

Figure 2. Example of warning dialog with an explanation message generated with APOLLO.

3.1. Preprocessor Module

The first module in the pipeline of APOLLO is the *preprocessor* module, which takes an email in .eml format in input and makes it more easily readable for the GPT model. This is done by following a similar approach to the preprocessing applied in the work of Misra and Rayz [35]. First, the subject email headers and body of the email are extracted; the headers are inserted in a dictionary, while all the HTML tags in the body (if any) are removed. Then, the information about any URL, phone number, and email address in the body is saved by applying a special meta-tag in place of the HTML anchor (`<a>`) tags. Finally, further preprocessing is performed to remove subsequent blanks or newline characters.

The preprocessing procedure achieves different goals. First, it reduces the length of the text to feed into the GPT model; this is essential since the context window is limited and might not be sufficient for treating a raw .eml file containing several headers, an HTML body, etc.; this is true especially if one decides to use a smaller GPT model. Another advantage of applying a preprocessing step is that of giving the LLM less textual information to work on, which could improve results [11].

At the end of the preprocessing step, the original email is broken down into: the dictionary of headers, the subject, the preprocessed body, and the list of URLs found in the email. The URLs will then be fed to the URL enricher module to extract additional information.

3.2. URL enricher Module

The *URL enricher* module is needed to retrieve threat intelligence to give the GPT model more grounded facts on which to reason. This approach is a form of Retrieval Augmented Generation [31], as it gathers information from external sources to improve the accuracy and reliability of the LLM model. In this version of APOLLO, we do not consider every URL retrieved in an email, but only the first one (in order of appearance). This decision brings some limitations to the tool but simplifies its development considerably. Therefore, from the first URL, the tool extracts the full hostname ("protocol://hostname", e.g. "https://www.google.com"), which we will call just URL for simplicity, and feeds it to 2 API services:

1. VirusTotal (https://www.virustotal.com), a very popular security tool for scanning files and URLs; feeding a URL into the API endpoint allows scanning it with more than 90 antivirus products and blocklists to produce a threat score for each of them. From the results of a scan, APOLLO takes the number of votes for the URL, which can be either *harmless*, *undetected*, or *malicious*.
2. BigDataCloud (https://www.bigdatacloud.com/ip-geolocation), an IP geolocation service to retrieve the server location of the website linked by a URL. From the URL, the IP address is extracted and used to retrieve a 3-character unique identifier of a country (e.g., "ESP" for Spain).

3.3. LLM prompter Module

The *LLM prompter* module finally takes the information obtained from the two previous steps and feeds them into the GPT model to produce a classification outcome for the email and an explanation. This is done by sequentially filling two prompts with the email data. The design of the prompts required considerable manual effort and several iterations to refine them according to best practices of prompt engineering and empirical observation of the outcomes for different inputs [15, 33, 41]. The prompts were also iteratively tested on a small subset of phishing and legitimate emails in the inbox of one of the authors; the outcomes were evaluated based on (i) whether the classification outcome was right or wrong, (ii) whether the phishing cues and the social engineering techniques indicated by GPT were meaningful (we did not want to force the model in producing hallucinated outputs), and (iii) whether we considered the generated explanations easy enough for non-experts to understand.

The first prompt makes GPT produce a JSON object that contains the result of the classification (phishing/legitimate and probability), a list of possible persuasion principles that were applied in the email, and a list of phishing (or legitimacy) indicators, each with an explanation. The prompt is then followed by the preprocessed email divided into its headers, subject, and body, and the URL information divided into geolocation information and VirusTotal data (i.e., the number of detectors that classified the URL as "harmless", "undetected", and "malicious").

GPT's response to the first prompt is used in a second one following a *prompt chaining* approach [14] to obtain a more refined explanation that could constitute an effective warning message, i.e., "Feature description + Hazard Explanation + Consequences of not complying with the warning" [4]. To obtain explanations that comply with that structure, we opted for a *few-shot prompting* approach [41] by including in our prompt three examples of such explanation messages manually designed by Desolda et al. for their study [16].

4. APOLLO's performance in phishing classification

4.1. Materials and Methods

To determine the performance of APOLLO in detecting phishing emails, we conducted an evaluation of the system in terms of accuracy in discerning between phishing and genuine emails. We tested the system by using a dataset of 4000 emails (half phishing, half genuine) sampled from the Nazario, NigerianFraud, and SpamAssassin datasets [12]. The sampling criteria were 1) emails with at least one link and ii) the most recent emails in the datasets. The final dataset used in our evaluation is available in the APOLLO repository. We used GPT-4o as the classification engine (specifically, version *gpt-4o-2024-05-13*), the most recent and performing OpenAI model for classification tasks at the time of writing this paper [38].

Emails were classified in sequence with a modified version of APOLLO. The "label" and "probability" fields were extracted from the outputs of GPT-4o: the former represents the predicted crisp label ("phishing"/"legitimate"), while the latter is the estimated probability of the email belonging to the "phishing" class, ranging from 0 (surely legitimate) to 1 (surely phishing).

An important aspect of the evaluation was assessing the impact of VirusTotal's URL information on classification performance. Since the emails in our dataset are over ten years old, VirusTotal can now label most URLs accurately, but this does not reflect real-time behavior during new phishing attacks, where URL classifications can take hours or days to stabilize [3, 43]. To simulate these varying accuracy levels, we analyzed VirusTotal's outputs for 4000 emails and an additional 4000 URLs from the PhishTank repository, identifying the following parameter ranges: $n_{harmless}$ = [0–87], $n_{undetected}$ = [0–28], and $n_{malicious}$ = [0–25]. These ranges show, for example, that the maximum number of VirusTotal detectors for malicious URLs is 25.

The evaluation began by testing GPT-4o's performance without URL information. Then, we simulated VirusTotal outputs at different confidence levels (e.g., Q100 for highly reliable outputs and Q0 for entirely uncertain data). We also tested APOLLO when considering misleading information to simulate the cases in which misleading VirusTotal information was fed to GPT-4o, i.e., phishing emails partially or entirely misclassified as legitimate and vice versa (e.g., Q25ERR to Q100ERR conditions). Even if this is a rare situation, given the high quality of this service, it is interesting to analyze how GPT-4o classifies emails when the prompt contains misleading information.

Geolocation data from BigDataCloud was included in all conditions using URL information. However, the individual effects of VirusTotal and BigDataCloud were not isolated, a limitation to address in future work. To limit the fluctuation in the results of the GPT model and obtain more deterministic outputs, we set the TEMPERATURE parameter in the LLM prompter module to a very low value (0.0001) throughout the evaluation process [37].

4.2. Data analysis

Precision, recall, accuracy, and F1-score are used to measure the model's performance in the email binary classification task. Moreover, log-loss and ROC AUC were used to measure the performance in the probability estimate for the two classes.

The chi-square test of independence was employed as an omnibus test to verify any differences in the predicted labels (dichotomous nominal value) among various conditions. Subsequently, the chi-square test was employed as a post-hoc test with Bonferroni correction in case of significant differences. Similarly, the One-way ANOVA was performed to assess any differences in the predicted probability; Tukey's HSD post-hoc tests were performed in case of significant differences.

4.3. Experimental results

The evaluation results show that the performance of GPT-4o in the email phishing detection task is very high, even without including additional URL information. Indeed, it resulted in a precision of 0.964, a recall of 0.985, and an accuracy of 0.974, indicating that the model can effectively distinguish between phishing and legitimate emails based solely on the email content and metadata; moreover, the low log-loss (0.113) and high ROC AUC (0.994) further confirm that the model provides reliable probability estimates and maintains excellent discriminative power (see **Table 1**).

A chi-square test of independence was performed to examine the relation between the different conditions in the cases where the right URL information is present or not. The relation between the examined conditions (noURL, Q0, Q25, Q50, Q75, Q100) and the predicted label was significant (X^5 = 653.39, p < .001). Thus, post-hoc tests were performed among all the pairs. The results of the pairwise comparisons indicated that by including the information on the URL when the attack starts (Q0), the accuracy decreases with respect to noURL (p <.001). This result shows how insignificant information returned from a third-party service (e.g. $n_{harmless}$=0, $n_{malicious}$=0) can confuse the classification of the model, making it worse than it would be without such information; we can conclude that in case of such information on the URL, the parameters and their values should be excluded from the prompt. The benefits of URL information begin to become apparent in the next quartiles. Indeed, in the case of Q25, the model's performance is almost perfect (F1-score=0.999), outperforming both the noURL (p = <.001) and Q0 (p = <.001) conditions. The same level of performance was observed in the case of Q50, Q75, and Q100, as in these instances, the model incorrectly classified only four emails.

Table 1. Performances of APOLLO in the binary classification task. The performance of GPT-4o becomes increasingly precise as the information about the URL becomes more certain (Q≥25) to a maximum at Q100, while adding uncertain information (Q0) decreases performance.

Condition	Prediction		Classification metrics				Regression metrics	
	Correct	Wrong	precision	recall	accuracy	F1	log-loss	ROC AUC
noURL	3896	104	0.964	0.985	0.974	0.974	0.113	0.994
Q0	3797	203	0.997	0.901	0.949	0.947	0.279	0.981
Q25	3996	4	0.998	1.000	0.999	0.999	0.804	0.962
Q50	3996	4	0.998	1.000	0.999	0.999	0.753	0.968
Q75	3996	4	0.998	1.000	0.999	0.999	0.254	0.992
Q100	3996	4	0.998	1.000	0.999	0.999	0.136	0.997
$Q25_{ERR}$	1795	2205	0.473	0.898	0.449	0.619	1.663	0.356
$Q50_{ERR}$	1775	2225	0.470	0.888	0.444	0.615	1.911	0.336
$Q75_{ERR}$	1790	2210	0.472	0.895	0.448	0.618	4.686	0.267
$Q100_{ERR}$	1798	2202	0.473	0.899	0.450	0.620	15.230	0.027

Further interesting outcomes come from the evaluations of GPT-4o in the case of wrong information from VirusTotal. Results indicate that even a small error ($Q25_{ERR}$) of these services may significantly hinder the model's performance. Increasing the amount of error in the prompt ($Q50_{ERR}$, $Q75_{ERR}$, $Q100_{ERR}$) increases the error in the predicted class probabilities (i.e., the log loss), making the model increasingly confident in giving a wrong outcome. However, injecting wrong URL information mostly affects GPT-4o's precision rather than its recall; this means that GPT-4o adopts a cautious approach by default and is robust to the introduction of wrong information, detecting phishing emails accurately in any case.

5. User evaluation of APOLLO-generated warnings

In order to evaluate the qualities of warnings generated by APOLLO, we conducted an explorative user study to measure how users perceived these warnings in the context of an email client. This section presents the results of this study, plus a comparison of these results to state-of-the-art solutions. Four warnings were created using APOLLO; as a baseline we used the study of Desolda et al. [16], considering four state-of-the-art warnings: a *manual explanation* warning that includes explanations crafted by experts (M), and the warnings of Google Chrome (C), Mozilla Firefox (F) and Microsoft Edge (E), which do not provide any explanation. To compare our warnings with the baselines in the most rigorous way possible, we replicated the study reported in the paper [16] in terms of design, user recruitment, scenario, and metrics.

5.1. Study design and participants

We adopted a within-subjects design with the *warning* being the independent variable, and the 8 within-subject levels being the 4 warnings proposed in this study, plus the 4 warnings of [16] as a baseline. The baselines were chosen because the *manual explanation* proved to be a valuable and effective warning that includes explanations, similar to the warnings in our study, and because the other warnings are those commonly used by most users [19].

Our study consisted of an online survey hosted on the *Lime Survey* platform. Participants were exposed to 4 phishing emails, each supplied with a specific warning. To recruit participants, we used the online platform Prolific (https://www.prolific.co); 20 participants (9 M, 11 F, 0 NB) took part (age avg=30.85, sd=9.57). The study was anonymous and took 20 minutes to complete. Participants were paid at a rate of £10.50/hour. The study was approved by the Research Ethics Committee of King's College London (Ethical Clearance Reference Number: *MRA-23/24-40811*).

5.2. Instruments and Measurements

We used APOLLO to generate explanation messages for 4 phishing emails that were heavily inspired by real emails; one of these emails was a genuine email from Facebook and was used as a false positive in our study. We forced APOLLO to generate a (wrong) explanation (W3) based on a feature that was observable in the email but that was harmless. The generated explanations were then used to build 4 warnings, reported in Appendix A: we replicated the warning dialog of the manual explanation baseline and replaced the explanation message with the explanations generated by APOLLO. The original emails are available at this link.

Two questionnaires were administered: (i) a questionnaire originally proposed by Desolda et al. [16] (items listed in Appendix B) to measure different aspects of a warning dialog, both quantitative (understandability, familiarity, interest, perceived risk, and trust) and qualitative ones (reaction, confusing words, perceived meaning and action to take); and (ii) the System Causality Scale (SCS) [24], a questionnaire that quantitatively measures the perceived quality of explanations through 5-point Likert scales (items are listed in Appendix C). For each warning to which participants were exposed, they had to fill in both questionnaires, resulting in a total of 8 (2 questionnaires x 4 conditions) questionnaires answered by each participant (160 questionnaires in total).

5.3. Quantitative Results

To analyze the quantitative data, we performed pairwise comparisons of the 8 experimental conditions, which included the 4 warnings proposed in this paper and the 4 baselines. If the p-value was significant ($\alpha<0.05$), an effect size r was calculated [27] and categorized into three levels: small ($0 < r \leq 0.3$), medium ($0.3 < r \leq 0.5$), and large ($0.5 < r \leq 1$).

The results of the study for the quantitative data (items 3, 4, 5, 7, and 10 of the warning evaluation questionnaire) are summarized in . The Mann-Whitney U-Test was used to perform pairwise comparisons of the 8 experimental conditions, which included the 4 warnings proposed in this paper (W1, W2, W3, and W4) and the 4 baselines (M (manual explanation), C (Chrome), F (Firefox), and E (Edge)). No differences were found by comparing the warnings proposed in this study, so they were omitted. We also omitted the results of the baseline comparisons, as they were already reported in the original paper [16]. For the remainder of this section, all the statistical differences that are reported are always in favor of the four warnings proposed in this study.

Regarding *understandability* (item 3), all warnings resulted in very high scores (W1 $\bar{x}$ = 4.9, W2 $\bar{x}$ = 4.75, W3 $\bar{x}$ = 4.75, W4 $\bar{x}$ = 4.9) and statistical differences emerged when compared to all the baselines. Regarding *familiarity* (item 4), high scores were achieved for all four warnings (W1 $\bar{x}$ = 3.9, W2 $\bar{x}$ = 3.5, W3 $\bar{x}$ = 3.85, W4 $\bar{x}$ = 3.95), with no statistical difference. Similar results were obtained for *perceived risk* (item 7), with all four warnings obtaining very high scores (W1 $\bar{x}$ = 4.2, W2 $\bar{x}$ = 4.1, W3 $\bar{x}$ = 4.0, W4 $\bar{x}$ = 4.2), but with no statistical difference. Concerning *interest* (item 5), the results were very positive for all four warnings (W1 $\bar{x}$ = 1.35, W2 $\bar{x}$ = 1.85, W3 $\bar{x}$ = 1.75, W4 $\bar{x}$ = 1.6 – with opposite polarity). W1 was more interesting than all the baselines, W4 was better than F, and all our warnings were better than M. Finally, perceived *trust* (item 10) resulted in positive scores (W1 $\bar{x}$ = 4.25, W2 $\bar{x}$ = 4.05, W3 $\bar{x}$ = 3.8, W4 $\bar{x}$ = 4.25). Statistical differences emerged between W1 and C, W2 and M, and W4 with both C and M.

	3. Understandability				4. Familiarity				5. Interest				7. Risk Felt				10. Trust			
	M	C	F	E	M	C	F	E	M	C	F	E	M	C	F	E	M	C	F	E
W1	H	L	L	L					L	L	L	L	L					L		
W2	M	L	L	L					L								M			
W3	M	L	L	L					L											
W4	H	L	L	L					L		L						M	L		

Figure 3. Overview of statistical test results a: a green cell indicates a Low effect size, yellow denotes a Medium effect size and red indicates a high effect size. M = manual explanation, C = Chrome explanation, F = Firefox explanation, E = Edge explanation.

The effectiveness of the warning dialog is measured by item 8 (**Figure 4**). Results for this item evidently show that W1, W2, and W4 led to a higher percentage of users choosing the safest action ("to not continue to the website"), i.e., 76.2% for all the warnings. The only exception was the warning related to the false positive W3, which reported a lower percentage of users (61.9%) choosing the safest action. On the contrary, all baselines led to a lower percentage of users choosing the safest action, and a higher percentage of users choosing "to be careful while continuing to the website", especially in the case of the manual explanation M (not continue = 52.8%, be careful while continuing = 42.7%).

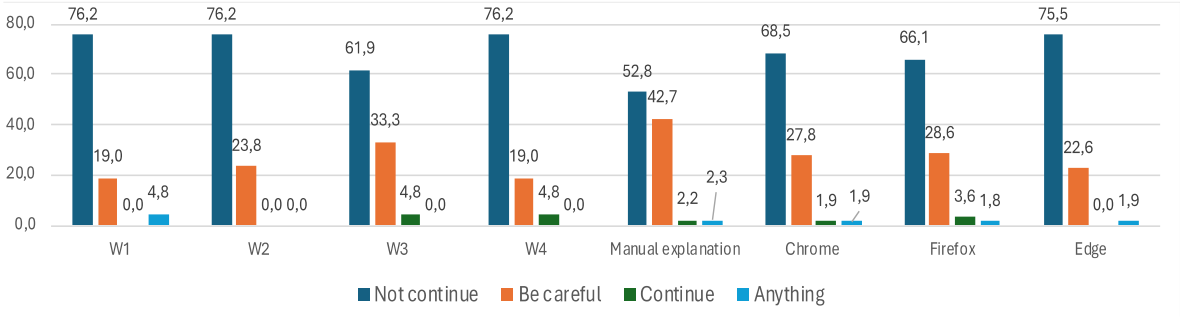

Figure 4. Percentages of each experimental condition for item 8

Regarding SCS, the score of all four warnings proposed in this study was high (W1 $\bar{x}$=.835, sd=.094; W2 $\bar{x}$=.804, sd=.131; W3 $\bar{x}$=.791, sd=0.112; W4 $\bar{x}$ =.829, sd=.098). The Mann-Whitney U-Test was applied to compare the scores of the SCS across our four experimental conditions (W1, W2, W3, W4). No significant differences emerged among the different conditions for both the individual items of the SCS and for the overall SCS scores.

5.4. Qualitative results

An inductive thematic analysis [6] of the qualitative data from items 2, 6, and 9 was performed by two authors individually to identify the major themes that would spontaneously emerge. From the analysis of answers to item 2 ("When you saw the warning dialog, what was your first reaction?") we identified 5 themes. The first theme, named "Emotive reaction" refers to users who reported an emotional response, such as feeling negative emotions (e.g., alert 11 times, confusion 4 times, panic 1 time, concern 8 times). The second theme, named "Stop the interaction – unsafe content", refers to users taking drastic action by interrupting the interaction due to unsafe content (26 occurrences). The third theme, named "Suspicious on the content", refers to users feeling that something dubious is happening (9 occurrences). The fourth theme, "Need to investigate", is linked to a state of uncertainty about the validity of the content, which leads users to want to investigate further before moving on (5 occurrences). The last theme, called "Nothing", relates to the lack of user reaction even after being exposed to a warning.

The analysis of the answers to item 9 ("What do you think this warning dialog means?") led to identifying three themes. The most common theme, named *"Generic Danger*", suggests that users interpreted the warning as referring to a potential threat to their personal information (26 occurrences), as well as to their devices (2 times), or a generic danger (10 times). The second theme, *"Phishing content"*, suggests that users perceive a definite risk of phishing. Within this theme, users perceived as dangerous the website to be opened (26 times), the email (2 times) or the link itself (5 times). The third and final theme called *"Potential unsafe content"* is similar to the previous one; however, it differs in the degree of certainty of the users. In this theme, in fact, we find responses from users who are unsure whether the content is phishing. The content perceived as potentially unsafe is, again, differentiated between the email (3 times), link (2 times) and website (11 times).

Finally, the results for item 6 report that some words in the explanations were reported as either confusing or too technical for the user, including "domain" (reported 7 times), "top-level" (5 times), "deceptive" (2 times), "IP address" (1 time).

6. Discussions

This study gave us some insights on how users perceive warning dialogs for phishing attacks generated by LLMs. Results suggest that **LLM-generated explanations show promise in protecting users.** In item 8, no users would engage in the most dangerous and unsafe behavior of continuing to the website. In fact, almost all participants chose the safest actions, i.e., "to not continue to the website" (76.2% of the time) and "be careful while continuing to the website" (19% of the time). These actions resulted in higher percentages than the baselines; notably, in the manual warning the safest action ("Not continue") was chosen by only 58% of participants.

Results also suggest that explanations generated by APOLLO are perceived as very **understandable,** outperforming warnings of Chrome, Firefox, and Edge with a low effect, and the manually written explanation with a medium (W2, W3) or high (W1, W4) effect. Warnings from APOLLO were also **interesting** for users, with W1 outperforming all the baselines, W4 outperforming Firefox, and all the warnings outperforming the manual explanation. Moreover, the positive results of the SCS questionnaire indicate that LLM-generated explanations are also perceived as **high-quality** overall. In fact, all of our warnings led to an SCS score of at least 0.79/1.00 (for W3).

The results related to the false positive email (W3) report that users had lower trust, suggesting that they could understand the warning's incorrectness thanks to the provided explanation. Moreover, users chose the safest action less frequently than with the other warnings. This suggests that **LLM-generated explanations may offer good support for identifying false positives**.

The data related to the SCS questionnaire (Items 1 and 3) suggests that there might be a need to **integrate the Need for Cognition** (NFC)[10] **into the design of warnings**. Users with high NFC may prefer more detailed explanations on demand in warnings, rather than superficial explanations, to understand why an email might be a scam or not. Implementing a customizable level of detail in warnings can accommodate both high and low NFC users.

During the analysis of item 2, which pertains to users' reactions, it was found that some users applied **emotional thinking** by reacting emotively, while others used rational thinking by stopping to reflect and investigate suspicious content. Discerning emotional from rational thinking is crucial as it may affect the response that users have to cyberattacks that trigger emotional responses such as fear or anxiety. Designing effective warnings requires finding the right balance between the two extremes, also considering user profiles.

7. Conclusions

In this paper, we presented APOLLO, a proof-of-concept of a unified system for both classifying phishing emails and generating warnings to alert the user. An assessment of the APOLLO's classification performance and of the user perception of its generated warnings was conducted. The results highlight the potential of LLM-based approaches for generating warnings for phishing emails, as it would also improve the scalability, and efficiency of the design process. However, it must be considered that explanations generated by an LLM are different from those generated by traditional methods (like eXplainable AI) and surely have several drawbacks [5, 8, 32, 42], such as hallucinations, explanations that do not reflect the model's internal processes, etc.

Future work includes repeating the evaluation study of APOLLO with other popular LLMs (e.g., Gemini 2.0, Llama3.3, etc.), rather than solely OpenAI's GPT-4o [38]. Moreover, by including many more users in a future quantitative user study it will be possible to measure the actual effectiveness of the LLM-generated warnings by means of metrics such as the click-through rate of users when exposed to the warning in a simulated scenario, in a setting similar to studies like [9, 39]. This will also allow us to compare LLM-generated warnings with manually generated messages and in terms of protecting users. This study can also include the assessment of LLM classification variability on the consistency and reliability of user interactions, which may require conducting multiple trials with varying inputs and carefully controlling for confounding variables to isolate the effects of LLM integration on user behavior.

Acknowledgements

This work has been supported by the Italian Ministry of University and Research (MUR) and by the European Union-NextGenerationEU, under grant PRIN 2022 PNRR "DAMOCLES: Detection And Mitigation Of Cyber attacks that exploit human vuLnerabilitiES" (Grant P2022FXP5B) CUP: H53D23008140001. This work is partially supported by the co-funding of the European Union - Next Generation EU: NRRP Initiative, Mission 4, Component 2, Investment 1.3 - Partnerships extended to universities, research centres, companies and research D.D. MUR n. 341 del 5.03.2022 – Next Generation EU (PE0000014 – "Security and Rights In the CyberSpace – SERICS" - CUP: H93C22000620001). The research of Francesco Greco is funded by a PhD fellowship within the framework of the Italian "D.M. n. 352, April 9, 2022"- under the National Recovery and Resilience Plan, Mission 4, Component 2, Investment 3.3 - PhD Project "Investigating XAI techniques to help user defend from phishing attacks", co-supported by "Auriga S.p.A." (CUP H91I22000410007).

Declaration on Generative AI

During the preparation of this work the authors used Grammarly and DeepL in order to improve language and readability. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. Akhawe, A. P. Felt. 2013. Alice in warningland: a large-scale field study of browser security warning effectiveness. Proc. USENIX conference on Security, SECURITY '13, (Aug 2013), 257–272, https://dl.acm.org/doi/10.5555/2534766.2534789

[2] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, E. Almomani. 2013. A Survey of Phishing Email Filtering Techniques. IEEE Commun. Surv. Tutor., 15(4), 2070-2090, https://ieeexplore.ieee.org/document/6489877

[3] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, K. Rieck. 2014. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. Proc. Symposium on Network and Distributed System Security, NDSS '14, (Feb 2014), http://dx.doi.org/10.14722/ndss.2014.23247

[4] L. Bauer, C. Bravo-Lillo, L. Cranor, E. Fragkaki. 2013. Warning Design Guidelines, url: https://www.cylab.cmu.edu/_files/pdfs/tech_reports/CMUCyLab13002.pdf

[5] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx *et al.* 2022. On the Opportunities and Risks of Foundation Models, url: http://arxiv.org/abs/2108.07258

[6] V. Braun, V. Clarke. 2006. Using thematic analysis in psychology. Qual. Res. Psychol., 3(2), 77-101, https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa

[7] C. Bravo-Lillo, L. F. Cranor, J. Downs, S. Komanduri, M. Sleeper. 2011. Improving Computer Security Dialogs. Proc. International Conference on Human-Computer Interaction, LNCS, INTERACT '11, (Sep 2011), 18-35, https://dl.acm.org/doi/10.5555/2042283.2042286

[8] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar *et al.* 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4, url: http://arxiv.org/abs/2303.12712

[9] P. Buono, G. Desolda, F. Greco, A. Piccinno. 2023. Let warnings interrupt the interaction and explain: designing and evaluating phishing email warnings. Proc. CHI Conference on Human Factors in Computing Systems, EA, CHI EA '23, (Apr 2023), 1-6, https://dl.acm.org/doi/abs/10.1145/3544549.3585802

[10] J. T. Cacioppo, R. E. Petty. 1982. The need for cognition. JPSP, 42(1), 116-131, https://psycnet.apa.org/record/1982-22487-001

[11] C. P. Chai. 2023. Comparison of text preprocessing methods. Nat. Lang. Eng., 29(3), 509-553, https://www.cambridge.org/core/product/43A20821D65F1C0C4366B126FC794AE3

[12] A. I. Champa, F. Rabbi, M. F. Zibran. 2024. Why Phishing Emails Escape Detection: A Closer Look at the Failure Points. Proc. International Symposium on Digital Forensics and Security, ISDFS, (29-30 April 2024), 1-6, https://ieeexplore.ieee.org/document/10527344
[13] DAIR.AI. Few-Shot Prompting - Prompt Engineering Guide. Retrieved 24 Jan. 2024 from https://www.promptingguide.ai/techniques/fewshot
[14] DAIR.AI. Prompt Chaining. Retrieved 25 Jul 2024 from https://www.promptingguide.ai/techniques/prompt_chaining
[15] DAIR.AI. Prompt Engineering Guide. Retrieved 24 Jan. 2024 from https://www.promptingguide.ai/
[16] G. Desolda, J. Aneke, C. Ardito, R. Lanzilotti, M. F. Costabile. 2023. Explanations in warning dialogs to help users defend against phishing attacks. Int. J. Hum.-Comput. Stud. 176, C, Article 103056 (Aug 2023), 20 pages. https://www.sciencedirect.com/science/article/pii/S1071581923000654
[17] G. Desolda, L. S. Ferro, A. Marrella, T. Catarci, M. F. Costabile. 2021. Human Factors in Phishing Attacks: A Systematic Literature Review. ACM Comput. Surv. 54, 8, Article 173 (Oct 2021), 35 pages. https://doi.org/10.1145/3469886
[18] A. El Aassal, S. Baki, A. Das, R. M. Verma. 2020. An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs. An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs, 8, 22170-22192, https://ieeexplore.ieee.org/document/8970564
[19] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, D. Wagner. 2012. Android permissions: user attention, comprehension, and behavior. Proc., SOUPS '12, (Jul 2012), Article 14 pages, https://doi.org/10.1145/2335356.2335360
[20] Z. Ghahramani. 2023. Introducing PaLM 2. (12 Sep 2023). Retrieved 20 Jan. 2024 from https://blog.google/technology/ai/google-palm-2-ai-large-language-model/
[21] P. M. Gholampour, R. M. Verma. 2023. Adversarial Robustness of Phishing Email Detection Models. Proc. ACM International Workshop on Security and Privacy Analytics, IWSPA '23, (Apr 2023), 67–76, https://doi.org/10.1145/3579987.3586567
[22] F. Greco, G. Desolda, A. Esposito. 2023. Explaining Phishing Attacks: An XAI Approach to Enhance User Awareness and Trust. Proc. The Italian Conference on CyberSecurity, 3488, ITASEC'23, (03-05 May 2023), https://ceur-ws.org/Vol-3488/paper22.pdf
[23] F. Heiding, B. Schneier, A. Vishwanath, J. Bernstein, P. S. Park. 2023. Devising and Detecting Phishing: Large Language Models vs. Smaller Human Models, url: https://doi.org/10.48550/arXiv.2308.12287
[24] A. Holzinger, A. Carrington, H. Müller. 2020. Measuring the Quality of Explanations: The System Causability Scale (SCS). KI, 34(2), 193-198, https://doi.org/10.1007/s13218-020-00636-z
[25] HuggingFace. Open LLM Leaderboard. 2024. Retrieved 13 Feb. 2024 from https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard
[26] IBM. 2024. Security X-Force Threat Intelligence Index. (21 Feb. 2024). Retrieved 14 May 2024 from https://www.ibm.com/reports/threat-intelligence
[27] K. Kelley, K. J. Preacher. 2012. On effect size. Psychol. Methods, 17(2), 137-152, https://psycnet.apa.org/record/2012-10789-001
[28] M. Khonji, Y. Iraqi, A. Jones. 2013. Phishing Detection: A Literature Survey. IEEE Commun. Surv. Tutor., 15(4), 2091-2121, https://ieeexplore.ieee.org/document/6497928
[29] T. Koide, N. Fukushi, H. Nakano, D. Chiba. 2023. Detecting Phishing Sites Using ChatGPT, url: https://arxiv.org/abs/2306.05816
[30] P. Kumaraguru, S. Sheng, A. Acquisti, L. F. Cranor, J. Hong. 2010. Teaching Johnny not to fall for phish. Trans. Internet Technol., 10(2), 1-31, https://doi.org/10.1145/1754393.1754396
[31] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal *et al.* 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, url: https://doi.org/10.48550/arXiv.2005.11401
[32] Q. V. Liao, J. W. Vaughan. 2023. AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap, url: https://arxiv.org/abs/2306.01941

[33] P. Liu, Q. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Comput. Surv. 55, 9, Article 195 (Sep 2023), 35 pages. https://doi.org/10.1145/3560815
[34] I. A. Marin, P. Burda, N. Zannone, L. Allodi. 2023. The Influence of Human Factors on the Intention to Report Phishing Emails. Proc., CHI '23, (Apr 2023), Article pages, https://doi.org/10.1145/3544548.3580985
[35] K. Misra, J. T. Rayz. 2022. LMs go Phishing: Adapting Pre-trained Language Models to Detect Phishing Emails. Proc. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT '22, (Nov 2022), 135-142, https://ieeexplore.ieee.org/document/10101955
[36] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature, url: http://arxiv.org/abs/2301.11305
[37] OpenAI. API Reference - OpenAI API. 2023. Retrieved 11 Jun 2024 from https://platform.openai.com/docs/api-reference/introduction
[38] OpenAI. Hello GPT-4o. 2024. Retrieved 12 Jul 2024 from https://openai.com/index/hello-gpt-4o/
[39] J. Petelka, Y. Zou, F. Schaub. 2019. Put Your Warning Where Your Link Is: Improving and Evaluating Email Phishing Warnings. Proc., CHI '19, (May 2019), Article pages, https://doi.org/10.1145/3290605.3300748
[40] S. Saxena. 2018. Precision vs Recall. (11 May 2018). Retrieved 13 Feb. 2024 from https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488
[41] J. Shieh. Best practices for prompt engineering with OpenAI API. (Jan 2024). Retrieved 21 Jan. 2024 from https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api
[42] M. Turpin, J. Michael, E. Perez, S. R. Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting, url: https://arxiv.org/abs/2305.04388
[43] M. Vasilescu, L. Gheorghe, N. Tapus. 2014. Practical malware analysis based on sandboxing. Proc. RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, RoEduNet & RENAM, (11-13 Sept. 2014), 1-6, https://ieeexplore.ieee.org/document/6955304
[44] G. Vilone, L. Longo. 2021. Notions of explainability and evaluation approaches for explainable artificial intelligence. Inf Fusion, 76(C), 89-106, https://doi.org/10.1016/j.inffus.2021.05.009

Appendix A. Explanation messages for each condition.

Warning	Explanation Message	Explained feature
W1	*The email contains a link that is just a string of numbers (an IP address). Legitimate companies usually use a name, not numbers. Clicking on it could lead to a fake site that steals your information.*	URL is an IP address
W2	*The email's URL has a top-level domain like '.com' oddly placed before another domain. This is a trick to make a malicious link seem trustworthy. If clicked, it could lead to a fake site that steals your personal details.*	Top-Level domain in the URL is mispositioned
W3	*The email shows a link labeled "protect your account", but it points to a different, suspicious website. This mismatch can trick you into visiting a harmful website. You might unknowingly give away your personal information or passwords*	Shown and actual links mismatch
W4	*The URL in the email leads to a very new domain. New domains are often used by scammers for fraud. You could be tricked into giving away personal details or downloading harmful software.*	Newly created domain

Appendix B. The questionnaire used to evaluate the warnings (from Desolda et al. [16])

#	Questionnaire item	Possible Answers

1	Did you read the entire text of the warning dialog?	[yes; partially; no]
2	When you saw the warning dialog, what was your first reaction?	[free text]
3	I understood the warning dialog	[5-point Likert scale, from “Strongly disagree” to “Strongly agree”]
4	I am familiar with this warning dialog	[5-point Likert scale, from “Strongly disagree” to “Strongly agree”]
5	I am not interested in this warning dialog	[5-point Likert scale, from “Strongly disagree” to “Strongly agree”]
6	Which word(s) did you find confusing or too technical?	[free text]
7	Please rate the extent of risk you feel you were warned about	[very low risk; low risk; no risk; risky; very high risk]
8	What action, if any, did the warning dialog want you to take?	[to continue to the website; to be careful while continuing to the website; to not continue to the website; I did not feel anything]
9	What do you think this warning dialog means?	[free text]
10	Please rate your level of trust in this warning dialog	[not at all confident; not very confident; neutral; confident; very confident]
11	What is the first word in this warning dialog?	[free text]

Appendix C. SCS questionnaire items (from Holzinger et al. [24])

#	**Questionnaire item**
1	I found that the data included all relevant known causal factors with sufficient precision and granularity
2	I understood the explanations within the context of my work
3	I could change the level of detail on demand
4	I did not need support to understand the explanations
5	I found the explanations helped me to understand causality
6	I was able to use the explanations with my knowledge base
7	I did not find inconsistencies between explanations
8	I think that most people would learn to understand the explanations very quickly
9	I did not need more references in the explanations: e.g., medical guidelines, regulations
10	I received the explanations in a timely and efficient manner