

Explaining Phishing Attacks: An XAI Approach to Enhance User Awareness and Trust

Francesco Greco¹, Giuseppe Desolda¹ and Andrea Esposito¹

¹University of Bari Aldo Moro, Bari, Italy

Abstract

Phishing is a cyber-attack that is a plague in today's digital society. AI solutions are already being used to detect phishing emails, but they typically do not address the problem of explaining to users why certain emails are considered dangerous. This leads to users not understanding the risk and/or not trusting the defense system, resulting in higher success rates of phishing attacks. This paper presents an XAI-based solution to classify phishing emails and alert users to the risk by explaining the reasons behind the attacks. We compared different ML models using a subset of features that can be explained and understood by non-IT users. We found that Explainable Boosting Machine was the best choice for a high-performance and interpretable classifier for email phishing detection.

Keywords

Phishing, Warning Dialogs, Explainable Artificial Intelligence, Human-Computer Interaction

1. Introduction


As technology advances and more of our lives take place online, we are increasingly getting exposed to cybercrime. In particular, phishing is one of the current biggest cyber threats, being the top infection vector for gaining initial access to the victims' network [1]. Phishing is a method used by criminals to steal personal information through fraudulent websites, emails, and phone calls. Since 2020, especially due to the COVID-19 pandemic and the shift to remote work, phishing attacks have increased significantly [2].


To combat phishing, many detection methods are used, although the most effective are those based on Artificial Intelligence (AI) [3, 4]. These AI models can detect suspicious emails and websites with very high precision (e.g., Google claims to have an anti-spam filter that catches 99.9% of spam and phishing emails [5]). Nonetheless, automatizing the phishing detection task would inevitably lead to misclassifications, even with state-of-the-art models: this behavior can easily compromise the user's productivity by blocking or deleting important emails [6]. Therefore, the final decision to access them has to be left up to the user. This can happen by showing warning dialogs that alert users about the dangerous nature of an email or website. However, attackers exploit human factors like stress and fear, and this leads users to bypass warnings [7, 2]. Moreover, not all users have the expertise or time to analyze the malicious

ITASEC 2023: The Italian Conference on CyberSecurity, May 03–05, 2023, Bari, Italy

✉ francesco.greco@uniba.it (F. Greco); giuseppe.desolda@uniba.it (G. Desolda); andrea.esposito@uniba.it (A. Esposito)

ORCID 0000-0003-2730-7697 (F. Greco); 0000-0001-9894-2116 (G. Desolda); 0000-0002-9536-3087 (A. Esposito)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

content on their own, and many fall victim to phishing attacks. These problems are amplified by the fact that such dialogs are designed without properly taking into account the users [8].

To improve the effectiveness of anti-phishing warnings, the user should understand the reasons why the system considers a message or website suspicious [9]. This means that warning dialogs should explain to the user the rationale behind the decision of the AI system. While AI models keep on making progress in terms of accuracy and performance thanks to technology and research innovation, they often remain hard or even impossible to be interpreted by a human. Predictors like these are black-box models since their functioning remains uninterpretable. This results in warnings that lack explanations and that do not provide insight into why the model considers the message or website malicious, leading to warnings that are less effective in conveying the danger of a phishing attack. To address the problem of “opening the black box”, the field of eXplainable Artificial Intelligence (XAI) has explored and developed several methods [10]. With XAI, it is possible to receive insight into the model’s reasoning process, but in a way that is aimed to be interpreted often only by AI specialists. Therefore, the outputs of an XAI tool should be adapted for the end-user to understand them, even with low or no technical knowledge.

In this paper, we present an approach that aims at increasing the effectiveness of warning dialogs for guarding users against phishing attacks by employing existing XAI methodologies for the phishing detection task. In particular, we developed an XAI-based tool that not only classifies emails as phishing/non-phishing but that also warns the end-user and explains to them the reasons why an email is considered dangerous. To do this, we followed a human-centered approach by considering aspects that are strictly related to users within the development process of the underlying AI system. The resulting warning messages have proven to be more effective than the state-of-the-art approach in a user study with 300 participants [11].

2. Related Work

To defend users from phishing, warnings have been vastly employed and improved in the years [12, 13, 14, 15]. Numerous design guidelines have to be considered when designing a warning dialog [16]. For example, to be effective a warning should be “active”, i.e., interrupt the user’s interaction flow and protect the user even when they do not read or understand the warning message [15]. Another finding coming from the literature regards the psychological effect of habituation [13]: users tend to ignore warnings if they look the same in different situations. To address this effect and improve their effectiveness, warnings should change their aspect based on the actual risk [17]; warnings that follow this principle are called *polymorphic*.

To improve the effectiveness of warning dialogs, users should understand the hazard and be motivated to heed the warning [9]. Explanations about why a particular phishing email can be dangerous can help the users evaluate the risk and make the correct choice. Moreover, when the decision is made by an AI agent, explanations can help increase the user’s trust in the system.

AI models can be very complex and not interpretable in their predictions; in this case, they are called “*black box*” models [10]. For example, deep neural networks are very hard to be made sense of, even for AI specialists. To interpret the decisions of an AI model and be able to generate an explanation understandable also by lay users, Explainable Artificial Intelligence

(XAI) approaches are fundamental. XAI can help obtain AI systems that provide clear and understandable explanations for their decisions. In particular, model-agnostic explanation tools can be applied to already existing machine learning models in a *post-hoc* manner [10]. Post-hoc XAI models are used to explain the output of an AI model related to individual instances. In the case of phishing detection, this means explaining the importance that the different features of, e.g., an email had on the final classification outcome. Examples of post-hoc XAI models are LIME (Local Interpretable Model-agnostic Explanations) [18] and SHAP (SHapley Additive exPlanations) [19], which can locally approximate any machine learning model to interpret the results by analyzing the input features and their impact on the model's predictions. Another approach is training AI models that are explainable by design and do not need post-hoc tools to be interpreted. Examples of these are linear models, decision trees, and rule-based models [20]. Explainable Boosting Machine (EBM) is an example of a model explainable by design, as it proposes highly explainable modeling to construct a prediction model which is explainable both locally and globally [21].

An explanation can be either *global* or *local*. A global explanation requires explaining the entire model and its functioning in general. A local explanation, instead, refers to explaining the outcome for an individual instance in particular. A local explanation can consist of a feature importance vector, i.e., a list of values that reports, for each input feature of an AI model, a numeric value that represents the importance of the feature for the model's outcome. As one can guess, the outputs of an XAI model cannot be easily understood by lay users as they are.

Few works tried to employ XAI tools in the phishing detection field; e.g. in [22], the authors have applied LIME [18] and EBM [21] to classify and generate explanations for phishing URLs. Nonetheless, none of such works considers the user in the design process of their solutions. Therefore, a novel approach is required to fill this gap. For example, Lin et al. [23] present a model for phishing webpages detection called *Phishpedia*, which, without needing a training set, takes as input a URL and a target brand list describing legitimate brand logos and their web domains, to return the classification result. In this approach, the warning is shown only when the user opens the phishing website and it might be too late to warn them. The alert consists of bounding boxes drawn around the most important visual elements that led to the classification outcome, like a fake brand logo or an input form; the URL of the website is shown as well, together with a warning about eventual similarities to URLs of known brands. The work by Franchina et al. [24] defines an approach for phishing email detection that utilizes both the metadata describing the composition of the e-mail and its content. The results of the classification procedure are shown to the user, reporting how each characteristic impacted the final score. Kluge and Eckhardt [25] propose the design of an approach that is almost model agnostic to produce explanations in the shape of text highlights, namely by drawing the user's attention to the telltale signs of phishing in a suspicious e-mail. To ensure that the highlighted words (i.e., the explanation) represent indeed phishing cues, the original AI model has to classify them as suspicious, and they have also to be sufficient themselves for the classification of the entire e-mail; this means that replacing the remainder of the email with different words should have a negligible influence on the classification output.

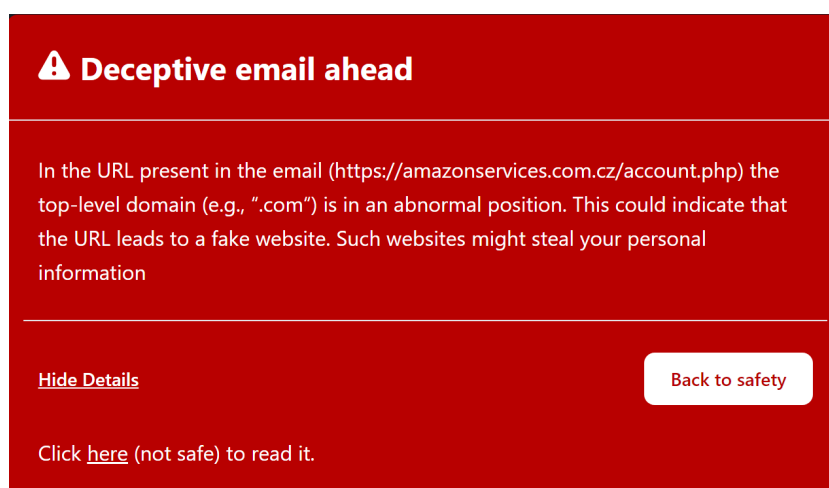


Figure 1: Proposed warning dialog

3. Designing a user-facing AI-based tool for phishing detection

Most current email clients use warnings to inform users that emails are suspicious. However, these warnings typically do not explain what exactly is suspicious. The lack of specific information puts the onus on the user to identify suspicious clues (such as the link). This increases the risk that warnings will be ignored or misunderstood. In addition, some features that indicate an email is phishing are invisible to the user, such as the age of the domain of the website linked to in the email, or its web ranking.

Our design approach focuses on the explanation process, starting from the phishing detection model, to increase the effectiveness of the warning. This solution extends the one recently presented by one of the authors of this paper, which mainly focuses on the design and evaluation of explanation messages in warning dialogs for phishing attacks [26]. This paper, instead, mainly focuses on the XAI models used to both classify phishing content and explain the outcome. The explanation is performed by computing a vector containing the linear contribution of each feature to the classification result. We used the feature importance as the basis for our user-facing explanations by associating each model feature with a human-readable message to be displayed in the warning.

In the design process of this tool, we also considered warning theory and guidelines that have been developed over the past decades (e.g., [16, 19, 27]). It is, however, out of the scope of this paper to report the detailed design process of the warning dialogs in our system. Figure 1 shows the designed warning dialog that is meant to appear when a user clicks on a link in a phishing email. The warning is “active” since it blocks the user’s interaction and forces their attention to its content [13]. It consists of a window with a title (on the top), an explanation message (in the middle), and two buttons (on the bottom); the “Back to Safety” button (in the bottom-right part of the window) makes the user return to the email client, the “Show details” button (in the bottom-left) extends the window with a section that contains a link (“here”) to follow the suspicious URL.

An important factor that we kept in mind when developing this tool is user habituation to warnings [28]. Repeated exposure to warnings can indeed result in habituation, which leads the users to ignore the warning even when there's a concrete danger. To mitigate habituation, we have designed our interface to behave in a polymorphic way [17], i.e., to show different explanations according to the phishing email feature. This is obtained on two different levels:

- The feature to give the user an explanation of why an email is likely to be phishing is randomly chosen between the 3 most relevant features in the decision made by the classification model.
- The actual explanation message of the single feature will randomly change between 2 different versions of the explanations.

4. Selecting the features set to train the XAI models

Before designing the warning messages, we first had to choose which features to explain to the user. These are part of the features adopted by the AI models to classify phishing emails. To avoid explaining features that cannot be easily understood by users with no IT knowledge (e.g., features that require specific knowledge of cybersecurity, networks, or web development), we have pursued the goal of generating a list of features that would be as intuitive as possible to be grasped by lay users. To discover what kind of features are generally used in phishing detection with machine learning, we have considered several works in the literature [3, 4]. We gathered over 140 features based on their usage in state-of-the-art solutions, reporting, for each of them, the name, the type (binary, discrete, continuous, or categorical), and a textual description; wherever the feature was not properly described in its relative work, we came up with a reasonable description on the best of our understanding, considering its name. Starting from these features, the authors of this paper manually analyze each of them, filtering the features that met two criteria:

1. features that are too costly to compute in terms of time; in fact, for the tool to be effective, we need to maximize Real-Time Application performance. For example, we excluded all the features that required making web requests to each of the websites pointed by the URLs in the email, wait for the different responses to arrive back, and then compute the features on the received response.
2. features that are too technical for a naïve user to make sense of, and for which we could not come up with a reasonable and concise explanation.

After the selection process, we came down to a total of 66 features. A lot of these had some concepts in common (e.g., “anomalies in the URL length” can be captured by counting the characters of the URL, the hostname or domain length, the length of the URL path, the average length of the domain tokens, etc.). Therefore, we conducted a grouping of features based on similarities between them, reaching a total of 18 features. The complete list of features is reported in Appendix Appendix A.

4.1. Design of the explanation messages for the warnings

As detailed at the beginning of this Section, the warning dialog we propose is characterized by an explanation message in the middle of the interface. A critical aspect of the design of the entire solution was how to explain to the users the technical concepts behind the feature chosen by the XAI model. To this aim, we designed, for each of the 18 features, two explanation messages following the design indications in the field of warnings for phishing attacks, such as the C-HIP model [29] and warning messages design guidelines [16]. In particular, we designed the messages to let them describe the risk comprehensively [30], and we followed a consistent layout [12]. The explanation messages follow a template which can be schematized as:

Description of the phishing feature + Hazard explanation + Consequences of a successful attack

An example of an explanation following this template, which explains the feature *Top-Level Domain mispositioned*, is the following:

In the URL present in the email the top-level domain is in an abnormal position. This could indicate that the URL leads to a fake website. Such websites might steal your personal information”.

The color coding of the sentences represents the role they have within the template. Two variants were created for each feature explanation to facilitate the creation of *polymorphic* warning dialogs, i.e., warnings that do not always look the same. Polymorphic warnings are required to limit the *habituation* effect [17]. Having numerous explanation messages allows us to create warnings that are very likely to differ one from another and, thus, limit the habituation effect. Therefore, we came up with 36 (2×18) different explanation messages in total. The complete list of explanation messages is reported in [11].

4.2. System design

The tool was designed to be placed within an email client and consists of different components. In Figure 2, a sequence diagram illustrates the functioning of the system and the user interaction flow. The user interacts with the *email client* and, whenever they open an email, the system computes both the classification output and the feature importance vector for that specific email. Based on the feature importance, a warning dialog (as seen in Figure 1) is generated and shown to the user, who can decide to heed it or ignore it and access the suspicious email.

The XAI system was designed to detect phishing emails starting from a raw HTML file, which represents the email to classify; the feature extractor component computes the 18 input features that are needed by the AI model (this is different from feature extraction as commonly intended in machine learning). We trained different AI models to perform a binary classification task, classifying an email as either “phishing” or “legitimate”. The selection of these models took into account the most adopted models in this field [3, 31, 4]. Among the machine learning models that were trained, some of them are interpretable by design, i.e., decision tree, logistic regression, and Explainable Boosting Machine (EBM) [21]. Conversely, others are black-box models, i.e., Support Vector Machine (SVM), Random Forest, Multi-Layer Perceptron (MLP),

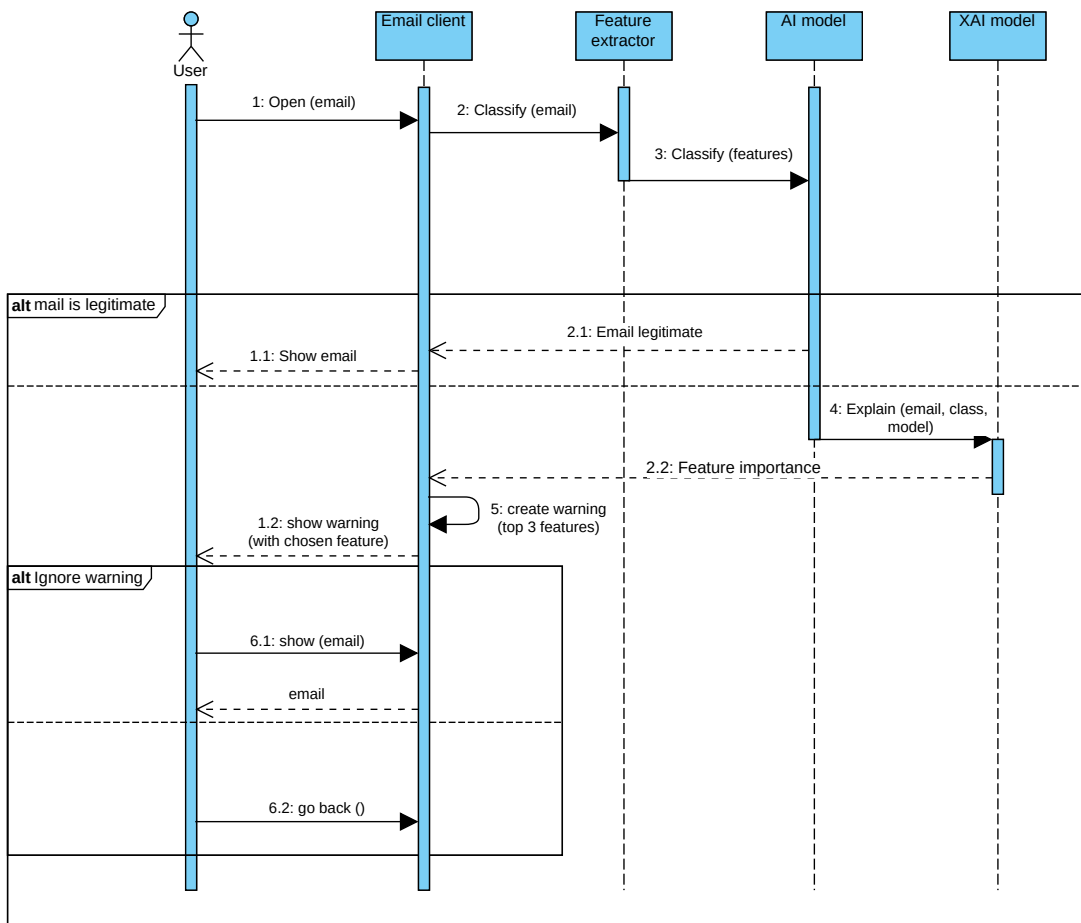


Figure 2: Sequence diagram of the system

Deep Neural Network (DNN). Therefore, to make these interpretable, we applied post-hoc XAI models, specifically LIME [18] and SHAP [19].

In case of the email being phishing, the *XAI model* (e.g., LIME or SHAP) takes the features of the email, the classification outcome of the AI model, and the model itself, and produces an explanation in the form of a feature importance vector. The latter is then sent to the email client, which creates the explanation message to show in the warning. To generate a *polymorphic* warning dialog, the system selects one of the explanation messages among those described in Section 4.1; the message is selected depending on the features that impacted the AI model’s classification outcome the most. Starting from the feature importance vector, the 3 most impactful features are candidates to constitute an explanation in the warning dialog. One of the 3 features is chosen at random and then one of its two versions is randomly selected.

5. Method

The data used to train the AI models originates from two different datasets: 2319 legitimate emails from the *SpamAssassin* dataset [32], and 1452 phishing emails plus 643 legitimate emails from the *Enron corpus* [33]. It is worth noting that these numbers come after a filtering process to delete emails that did not include at least one link in them. This is necessary to conform to our assumptions of phishing emails that should have at least one phishing link. With a total of 2962 legitimate emails and 1452 phishing emails, we obtain a ratio of about 2 : 1. We decided to apply an 80-20 split for the training and testing set with 5-fold cross-validation [34], to mitigate overfitting and selection bias effects. This is done in a stratified fashion so that the same legitimate-phishing ratio is kept for both training and testing folds. As anticipated in Section 4.2, a *feature extractor* component preprocesses the raw email files and produces an array of 18 features for each one (see Appendix A).

In an email, the feature extractor computes the features related to a URL (e.g., *self-signed HTTPS certificate*, *Top-Level Domain mispositioned*, *URL length*, etc.) starting from a link, which is either retrieved from an anchor tag or just found in plain text. The script for the feature calculation computes the features related to the URL *for each* link found in the emails. This means that for an email we should have an array of N sets of features, one for each URL. To have the data in a consistent format, we decided to consider only the most “dangerous” URL in each email, based on the value of its features. Finally, all the features are scaled, with a *MinMax* normalization to have only values in a [0,1] range, and standardized.

Since we used model-agnostic explainers we were not tied to any model for the phishing detection task. The machine-learning (ML) models were chosen based on the performance in the literature [3, 31, 4]: decision tree, logistic regression, SVM, random forest, an MLP (with 2 intermediate fully-connected layers), a DNN (with 4 hidden layers and 1 dropout layer), and EBM. Python’s Scikit-learn [35] library was used to train the different machine-learning models. Keras library (in particular, Sequential model and Keras Tuner) [36] was used to build and train the neural networks, and to tune their hyperparameters.

For the hyperparameter tuning of every model, we applied a 5-fold stratified cross-validation, using a custom seed (= 42) to make the results repeatable. To find the locally optimal hyperparameters for the ML models, we performed the tuning using a grid search technique. For the two neural network models (MLP and DNN), instead, we used a random search [37]. In the end, we chose the model with the parameters that carried the best performance according to the *f1-score*.

To protect the user from receiving phishing emails without displaying a warning, it is crucial to minimize the number of False Negatives (FN). To find a good tradeoff between precision and recall, at first, we trained all the models with unbalanced class weights: the phishing class was set to have 5 times the weight of the legitimate class, in order to reduce the FN rate. We searched for the class weights that, for each model, optimized the performances. The optimal configurations for each model resulting from the hyperparameter tuning process are reported in Appendix B.

The models were tested on the whole dataset using 5-fold stratified cross-validation, measuring the average precision, recall, negative predicted value, specificity, accuracy, Area Under the Receiver Operating Characteristic Curve (ROC AUC) score, and f1-score, across the folds. The

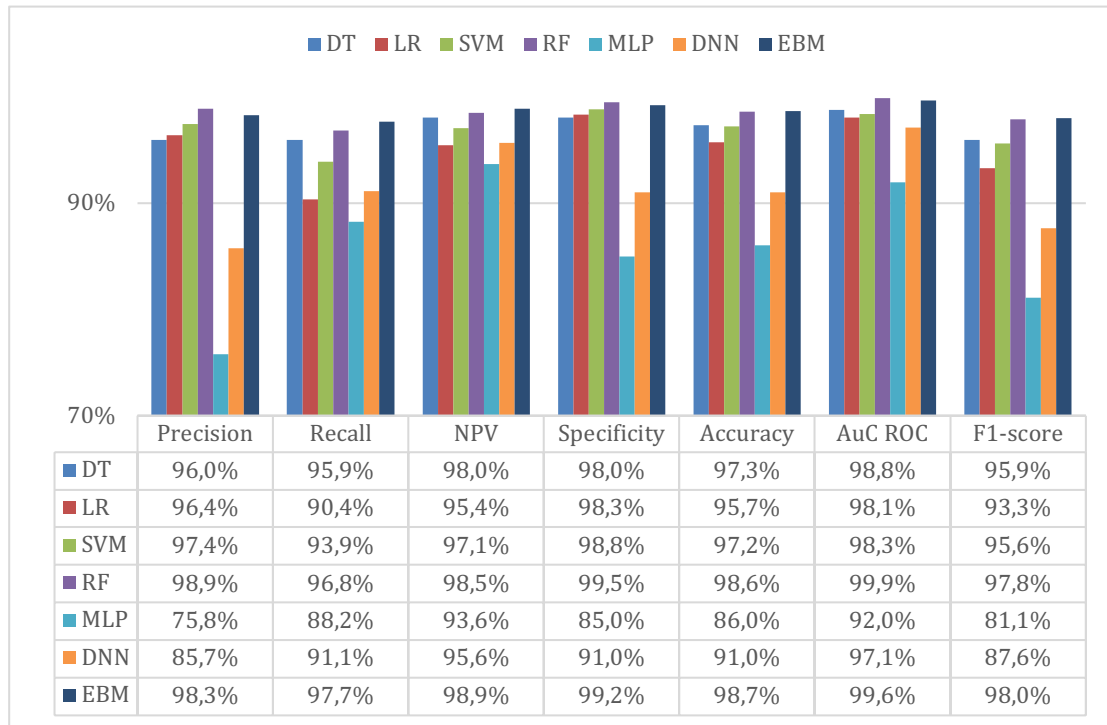


Figure 3: Classification results for all the models (DT: Decision tree, LR: Logistic Regression, SVM: Support Vector Machine, RF: Random Forest, MLP: Multi-Layer Perceptron, DNN: Deep Neural Network, EBM: Explainable Boosting Machine)

measurements for each model are reported in Figure 3. The results are reported as percentages rounded to the first decimal place. We can observe that the best performance is achieved by the Explainable Boosting Machine (EBM) and the Random Forest (RF) models, which achieve, respectively, 98.0% and 97.8% F1-score values. The model that performed the worst, instead, is MLP, with an F1-score of 81.1%.

To warn the user, the email client needs 3 features to generate an explanation in a polymorphic manner by randomly choosing a feature to show in the warning dialog, as discussed in Section 4.2. The explanations from the XAI tool are generated in different ways, depending on the underlying ML model:

- The decision tree model is explained using the Gini impurity metric for each node of the tree that is found in the decision path. The Gini impurity of a node can tell us how the feature used in the node splitting was relevant in telling the two classes apart.
- EBM is defined as a “glass-box” model since it is explainable by design [21]. With EBM, the tool can easily obtain both the model explanation and instance explanations.
- All the other ML models are explained with two post-hoc XAI models, i.e., LIME and SHAP. LIME can be applied to all of the other XAI models to generate an explanation of an instance (i.e., a specific email). SHAP is a post-hoc XAI model analogous to LIME, but it is also able to quickly generate a model explanation.

6. Discussions and Conclusion

The performances of the models on the test set (Figure 3) are quite satisfying, especially if we consider that it is possible to interpret the behavior of the models. We want to limit as much as possible the number of phishing emails that may reach the user without the system displaying a warning. In other words, we should minimize the number of phishing emails that are misclassified as genuine, preferring XAI models with a low False Negative (FN) rate; i.e., we should favor *recall* over *precision*. Therefore, EBM, Random Forest, and Decision Tree are good candidates to constitute the phishing detection classifier. In particular, we can observe that the highest recall and F1-score are obtained with EBM, which is also interpretable by design.

Regarding the limitations of this work, the selection of the features to train the models has been conducted by our team based on common sense and personal knowledge. To shift more towards a human-centered approach, in future works we need to consider setting a user study to find which features best fit an explanation for the end-user. Another limitation regards the public dataset used for the training of the AI models, which dates back to 2008. The scarcity of publicly available email corpora is due to the existence of privacy-related issues in the publication of private data. Since phishing is a continuously evolving attack, we plan to gather a new dataset of real-world phishing emails coming from an IT company. Finally, with the assumption of phishing emails having at least one phishing link, we are neglecting attacks based on attachments and/or social engineering. Future work will include conducting a longitudinal user study to assess both the in-vivo performance of the tool and the effect of the polymorphic warnings on habituation. Despite the efforts in research, phishing continues to be a critical problem, and the number of victims is increasing year after year. This work aims at increasing the user's trust in security software by giving them explanations about the causes of why specific emails are dangerous. We designed and developed a high-performant AI-based tool, which can help users correctly detect phishing emails, giving them the ability to make informed decisions, as it resulted in a user study with 300 participants [11].

Acknowledgments

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). The research of Andrea Esposito is funded by a Ph.D. 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 - Ph.D. Project “Human-Centered Artificial Intelligence (HCAI) techniques for supporting end users interacting with AI systems”, co-supported by “Eusoft S.r.l.” (CUP H91I22000410007). This Publication was produced with 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).

References

- [1] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you? : Explaining the predictions of any classifier, 2016. URL: <https://doi.org/10.1145/2939672.2939778>. doi:10 . 1145/2939672 . 2939778.
- [2] M. Wogalter, Communication-Human Information Processing (C-HIP) Model, 1st edition ed., CRC Press, 2018, pp. 33–49. doi:10 . 1201/9780429462269-3.
- [3] IBM, X-force threat intelligence index, 2022. URL: <https://www.ibm.com/downloads/cas/ADLMYLAZ>.
- [4] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 4768–4777.
- [5] N. Kumaran, Understanding gmail’s spam filters, 2022. URL: <https://workspace.google.com/blog/identity-and-security/an-overview-of-gmails-spam-filters>.
- [6] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, *ACM Computing Survey* 51 (2018) 42. URL: <https://doi.org/10.1145/3236009>. doi:10 . 1145/3236009.
- [7] M. Wu, R. C. Miller, S. L. Garfinkel, Do security toolbars actually prevent phishing attacks?, in: SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2006, p. 601–610. URL: <https://doi.org/10.1145/1124772.1124863>. doi:10 . 1145/1124772 . 1124863.
- [8] C. Bravo-Lillo, L. F. Cranor, J. Downs, S. Komanduri, M. Sleeper, Improving computer security dialogs, in: International Conference on Human-Computer Interaction, volume LNCS of *Human-Computer Interaction*, Springer Berlin Heidelberg, 2011, pp. 18–35.
- [9] J. Ellis, Covid-19 phishing update: Campaigns exploiting hope for a cure, 2020. URL: <https://info.phishlabs.com/blog/covid-phishing-update-campaigns-addressing-a-cure>.
- [10] B. B. Anderson, C. B. Kirwan, J. L. Jenkins, D. Eargle, S. Howard, A. Vance, How polymorphic warnings reduce habituation in the brain: Insights from an fmri study, 2015. URL: <https://doi.org/10.1145/2702123.2702322>. doi:10 . 1145/2702123 . 2702322.
- [11] P. Buono, G. Desolda, F. Greco, A. Piccinno, Let warnings interrupt the interaction and explain: Designing and evaluating phishing email warnings, in: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems, CHI EA '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 6. URL: <https://doi.org/10.1145/3544549.3585802>. doi:10 . 1145/3544549 . 3585802.
- [12] G. Desolda, L. S. Ferro, A. Marrella, T. Catarci, M. F. Costabile, Human factors in phishing attacks: A systematic literature review, *ACM Computing Survey* 54 (2021) 35. URL: <https://doi.org/10.1145/3469886>. doi:10 . 1145/3469886.
- [13] S. Egelman, L. F. Cranor, J. Hong, You’ve been warned: An empirical study of the effectiveness of web browser phishing warnings, in: SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2008, p. 1065–1074. URL: <https://doi.org/10.1145/1357054.1357219>. doi:10 . 1145/1357054 . 1357219.
- [14] M. Wogalter, Purposes and scope of warnings, *Handbook of Warnings* (2006) 3–9.
- [15] F. Chollet, et al., Keras, 2015. URL: <https://keras.io>.
- [16] M. Khonji, Y. Iraqi, A. Jones, Phishing detection: A literature survey, *IEEE Communications*

- Surveys & Tutorials 15 (2013) 2091–2121. doi:10.1109/SURV.2013.032213.00009.
- [17] E. Montalbano, Top email protections fail in latest covid-19 phishing campaign, 2020. URL: <https://threatpost.com/top-email-protections-fail-covid-19-phishing/154329/>.
- [18] A. El Aassal, S. Baki, A. Das, R. M. Verma, An in-depth benchmarking and evaluation of phishing detection research for security needs, *IEEE Access* 8 (2020) 22170–22192.
- [19] L. Franchina, S. Ferracci, F. Palmaro, Detecting phishing e-mails using text mining and features analysis, in: *Italian Conference on CyberSecurity*, volume 2940, CEUR-WS.org, 2021, p. 14.
- [20] K. Kluge, R. Eckhardt, Explaining the suspicion: Design of an xai-based user-focused anti-phishing measure, in: *International Conference on Wirtschaftsinformatik, Innovation Through Information Systems*, Springer International Publishing, 2021, pp. 247–261.
- [21] M. S. Wogalter, V. C. Conzola, T. L. Smith-Jackson, Research-based guidelines for warning design and evaluation, *Applied Ergonomics* 33 (2002) 219–230. URL: <https://www.sciencedirect.com/science/article/pii/S0003687002000091>. doi:[https://doi.org/10.1016/S0003-6870\(02\)00009-1](https://doi.org/10.1016/S0003-6870(02)00009-1).
- [22] L. Bauer, C. Bravo-Lillo, L. Cranor, E. Fragkaki, Warning design guidelines (cmu-cylab-13-002), 2013. URL: https://kithub.cmu.edu/articles/journal_contribution/Warning_Design_Guidelines_CMU-CyLab-13-002_/6468131.
- [23] Y. Lin, R. Liu, D. M. Divakaran, J. Ng, Q. Chan, Y. Lu, Y. Si, F. Zhang, J. Dong, Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages (usenix security 2021), 2021. URL: <https://www.usenix.org/conference/usenixsecurity21/presentation/lin>.
- [24] C. Bravo-Lillo, L. F. Cranor, J. Downs, S. Komanduri, Bridging the gap in computer security warnings: A mental model approach, *IEEE Security & Privacy* 9 (2011) 18–26. doi:10.1109/MSP.2010.198.
- [25] P. R. Galego Hernandes, C. P. Floret, K. F. Cardozo De Almeida, V. C. Da Silva, J. P. Papa, K. A. Pontara Da Costa, Phishing detection using url-based xai techniques, 2021. doi:10.1109/SSCI50451.2021.9659981.
- [26] G. Desolda, J. Aneke, C. Ardito, R. Lanzilotti, M. F. Costabile, Explanations in warning dialogs to help users defend against phishing attacks, *International Journal of Human-Computer Studies* 176 (2023) 103056. URL: <https://www.sciencedirect.com/science/article/pii/S1071581923000654>. doi:<https://doi.org/10.1016/j.ijhcs.2023.103056>.
- [27] 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, E. Duchesnay, Scikit-learn: Machine learning in python, *Journal of Machine Learning Research* (2011).
- [28] C. Molnar, *Interpretable Models*, Leanpub, 2020. URL: <https://christophm.github.io/interpretable-ml-book/simple.html>.
- [29] P. Rezaeilzadeh, L. Tang, H. Liu, *Cross-Validation*, Springer US, Boston, MA, 2009, pp. 532–538. URL: https://doi.org/10.1007/978-0-387-39940-9_565. doi:10.1007/978-0-387-39940-9_565.
- [30] B. Klimt, Y. Yang, *Introducing the enron corpus*, 2004.
- [31] J. Petelka, Y. Zou, F. Schaub, Put your warning where your link is: Improving and evaluating email phishing warnings, in: *Conference on Human Factors in Computing Systems*,

- ACM, New York, NY, USA, 2019, p. 1–15. URL: <https://doi.org/10.1145/3290605.3300748>. doi:10.1145/3290605.3300748.
- [32] A. Schwartz, SpamAssassin, O'Reilly Media Inc., 2004.
- [33] H. Nori, S. Jenkins, P. Koch, R. Caruana, Interpretml: A unified framework for machine learning interpretability, 2019. URL: <https://www.microsoft.com/en-us/research/publication/interpretml-a-unified-framework-for-machine-learning-interpretability/>.
- [34] C. Bravo-Lillo, L. F. Cranor, J. S. Downs, S. Komanduri, Bridging the gap in computer security warnings: A mental model approach, *IEEE Security & Privacy* 9 (2011) 18–26. doi:10.1109/MSP.2010.198.
- [35] S. Kim, M. Wogalter, Habituation, dishabituation, and recovery effects in visual warnings, *Human Factors and Ergonomics Society Annual Meeting Proceedings* 53 (2009) 1612–1616. doi:10.1518/107118109X12524444080675.
- [36] G. Desolda, F. Di Nocera, L. Ferro, R. Lanzilotti, P. Maggi, A. Marrella, Alerting users about phishing attacks, in: *International Conference on Human-Computer Interaction for Cybersecurity, Privacy and Trust*, volume LNCS of *HCI for Cybersecurity, Privacy and Trust*, Springer International Publishing, 2019, pp. 134–148.
- [37] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, E. Almomani, A survey of phishing email filtering techniques, *IEEE Communications Surveys & Tutorials* 15 (2013) 2070–2090. doi:10.1109/SURV.2013.030713.00020.

A. Appendix A

In Table 1 it is reported the complete list of 18 features for the AI models. The column Type tells the type of the values for the feature (B=Boolean, D=discrete number); the column Part of mail tells from which part of the email the feature is computed.

Table 1
List of Features

Feature name	Description	Type	Part of mail
Image Presence	A binary feature that is equal to 1 if there is a tag in the email body	B	Body
Links Present	Number of links in the e-mail body	D	Body
Misspelled Words	Number of misspelled words	D	Body
Special Characters in body	Occurrences of special characters in the email body	D	Body
Suspicious Words	One or more suspicious words are found in the e-mail body	D	Body
Age of Domain	The age of the domain (a site with less than 2 years is suspicious)	D	URL (external)
Expiration	The expiration date of the web domain	D	URL (external)
Ranking	The ranking of the website according to Google, Alexa	D	URL (external)
No HTTPS	The website does not use the HTTPS protocol	B	URL
Self-signed HTTPS certificate	The website uses HTTPS protocol with a not trusted certificate issuer (ex. self-signed certificates)	B	URL
Special Chars in URL	Presence of special characters (<code>_</code> , <code>/</code> , <code>//</code> , <code>@</code> , <code>-</code>), Unicode characters, or digits in the URL	D	URL
Link Mismatch	A binary feature that equals 1 if a link displayed in the email body is different than the redirected website	B	URL
Sensitive words in URL	Counts the number of sensitive words (i.e., “ <i>secure</i> ”, “ <i>account</i> ”, “ <i>webscr</i> ”, “ <i>login</i> ”, “ <i>ebayisapi</i> ”, “ <i>signin</i> ”, “ <i>banking</i> ”, “ <i>confirm</i> ”) in the URL	D	URL
IP address	There are one or more URLs with an IP Address as the domain	B	URL
TLD mispositioned	There is an anomaly in the position of the Top-Level Domain is in an abnormal position (it is either in the path of the URL or in one of the subdomains)	B	URL
Number of sub-domains	There is an anomaly in the domain of the URL since there are too many subdomains	D	URL
URL Length	There is an anomaly in the length of the entire URL or the length of the domain	D	URL
URL is shortened	A URL is shortened (e.g., TinyURL, etc.)	B	URL

B. Appendix B

In the following, the optimal configurations for the ML models are reported.

Decision Tree (DT)

- ccp_alpha: 0.0
- criterion: 'entropy'
- max_depth: 8
- min_sample_leaf: 7

Logistic Regression (LR)

- C: 100
- penalty: 'l2'
- solver: 'lbfgs'

Support Vector Machine (SVM)

- C: 100
- degree: 3
- gamma: 0.1
- kernel: 'poly'

Random Forest (RF)

- max_features: 5
- max_samples: 0.5
- n_estimators: 50

EBM No configurable hyperparameter.

Multi-Layer Perceptron (MLP)

- Class_weights: [Legit: 1, Phishing: 2]

Network configuration:

Layer (Type)	Output Shape	No. Parameters
Normalization (Normalization)	(None, 18)	3
dense_1 (Dense)	(None, 240)	4560
dense_2 (Dense)	(None, 240)	57840
output (Dense)	(None, 2)	482

- Total params: 62,885
- Trainable params: 62,882
- Non-trainable params: 3

Deep Neural Network (DNN)

- Class_weights: [Legit: 1, Phishing: 2]

Network configuration:

Layer (Type)	Output Shape	No. Parameters
Normalization (Normalization)	(None, 18)	3
dense_1 (Dense)	(None, 16)	304
dense_2 (Dense)	(None, 16)	272
dense_3 (Dense)	(None, 16)	272
dense_4 (Dense)	(None, 16)	272
dropout (Dropout)	(None, 16)	0
output (Dense)	(None, 2)	482

- Total params: 1,157
- Trainable params: 1,154
- Non-trainable params: 3