Explaining Suspected Phishing Attempts with Document Anchors

Kilian Kluge [1] **Regina Eckhardt** [1]

Abstract

Providing explanations for the decisions of machine-learning based phishing detectors appears as a promising means to prevent users from falling victim to phishing attacks. Building on prior work on explanations for document classification, we introduce *document anchors* that convey the key elements of a text that influence its classification by a black-box model. We demonstrate that the algorithm reliably extracts segments of text considered helpful for the discrimination between genuine and phishing e-mails.

1. Introduction

Phishing is a social engineering attack carried out over electronic communication channels. Attackers imitate a trustworthy source to gain confidential information from a user for malicious purposes (Kumaraguru et al., 2010; Pienta et al., 2018). Primarily executed via e-mail, increasingly personalized and sophisticated phishing attacks result in significant losses both for businesses and private individuals (Pienta et al., 2018; Williams et al., 2018).

Measures for phishing prevention are commonly divided into three categories: Blocking malicious e-mails before they reach users, warning users, and training users not to fall for phishing (Kumaraguru et al., 2010). Frequently employed measures of the first category include automated filters and machine-learning based phishing detectors. While state-of-the-art phishing detectors are robust and versatile, they suffer from their inherently limited accuracy. In order to prevent genuine e-mails from being rejected, phishing detectors are generally tuned for maximum precision at the expense of recall. Hence, many e-mails that were identified as suspicious of constituting a phishing attack reach the user (Albakry & Vaniea, 2018). Various studies found that between 10% and 30% of users that receive a phishing e-mail act on it (Williams et al., 2018; Pienta et al., 2018). Therefore, as the so-called second line of defense, user-focused anti-phishing measures aim to reduce users' susceptibility to phishing attempts. However, warnings and anti-phishing trainings are often found to be ineffective (Dennis & Minas, 2018; Kumaraguru et al., 2010). On the one hand, warning messages are frequently ignored by users (Kumaraguru et al., 2010). On the other hand, even users that were successfully trained to identify phishing e-mails nevertheless fall for phishing in everyday situations. This is attributed to a lack of users' awareness when performing routine tasks in a familiar and trusted environment (Williams et al., 2018; Dennis & Minas, 2018).

Against this background, approaches from the field of Explainable Artificial Intelligence (XAI) appear as a promising means to leverage the power of phishing detectors to design effective user-focused anti-phishing measures. In this context, Albakry & Vaniea (2018) envisioned an XAI system that explains to the user why an URL is suspected of leading to a fraudulent website. We take a broader approach and aim to direct the user's attention to any telltale signs in the text of suspicious e-mails by generating explanations for the output of a machine-learning based phishing detector. In contrast to generic warning messages, these explanations are specific to a particular suspicious e-mail and intended to both effectively raise users' awareness and guide their assessment. As a first step towards this goal, drawing from prior work by Ribeiro et al. (2018) and Lei et al. (2016), we introduce an algorithm for explaining document classification to lay users. We demonstrate that, combined with a deep learning phishing detector, the algorithm can uncover cues and phrases in e-mails that are relevant for identifying phishing attempts (cf. Parsons et al., 2016; Williams et al., 2018). Further, we provide evidence that the algorithm can be implemented efficiently for application at large scales.

The remainder of the paper is structured as follows: In section 2 we briefly review related research on explanations for document classification. Subsequently, in section 3, we introduce the *document anchors* algorithm. In section 4 we demonstrate its applicability using a real-world phishing dataset. We conclude the paper with a discussion of our results and an outlook on further research in section 5.

[1]Institute of Technology and Process Management, University of Ulm, Ulm, Germany. Correspondence to: Kilian Kluge <kilian.kluge@uni-ulm.de>.

2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020).

2. Related Work

A phishing detector is a binary classifier that discriminates between legitimate and phishing e-mails. Thus, explaining the output of a phishing detector is equivalent to explaining document classification. A variety of approaches have been proposed, which can be divided into search-based algorithms and document classifiers with integrated explanation capabilities.

Search-based algorithms are model-agnostic and thus applicable to any classifier. Martens & Provost (2014) focus on explanations for the classification of documents in bag-of-words representation. They define an *explanation* as a minimal set of words that, if removed, changes the classification. To find *explanations*, they utilize a best-first heuristic search with search tree pruning. In the case of a non-linear classifier, two post-processing optimizations aim to ensure that the found set is indeed minimal. Fernandez et al. (2019) generalize this approach to replacing words instead of removing them and introduce a variable cost for replacement, allowing for more fine-grained control of explanation properties. Similar to these *explanations*, the *anchors* introduced by Ribeiro et al. (2018) are sets of words. However, instead of constituting a minimal set of words required for the classification, *anchors* aim to be representative of the classifier. They are defined as a set of words that, if present, is sufficient to guarantee the classification independent of changes to the remainder of the document. *Anchors* are built up word by word through beam search. The KL-LUCB algorithm (Kaufmann & Kalyanakrishnan, 2013) is used to determine the best anchor candidates in each iteration.

Instead of generating explanations post hoc, Lei et al. (2016) train two joint models to find *rationales* for the classification of texts encoded as sequences of tokens. While an encoder model classifies a text, a generator model extracts the corresponding *rationales*, which are defined as short phrases that, individually, are classified similarly as the full text. An objective function ensures both correct classification and the *rationales'* conciseness and coherence. With their τ-SS3 classifier, Burdisso et al. (2019) pursue again a different approach. τ-SS3 is inherently interpretable, i.e., the classifier itself transparently reveals which word sequences in a text stream contributed most to its output. Further, the algorithm's design aims at enabling efficient implementation by relying only on basic data structures and functions.

3. Document Anchors

Users have to rely on cues to distinguish between genuine and phishing e-mails (cf. Table 1). Hence, the proposed anti-phishing XAI system should highlight suspicious elements of e-mails presented to users, drawing their attention to the most relevant words and phrases.

To be suitable for real-world application, the underlying XAI algorithm has to fulfill three requirements: First, its explanations should be comprehensible for lay users who have no technical knowledge concerning their generation (Ribeiro et al., 2018; Bhatt et al., 2020). Second, to be widely applicable and not interfere with the phishing detector, the explanation algorithm should be model-agnostic (cf. Ribeiro et al., 2018; Fernandez et al., 2019). Third, for application at large scales, the algorithm's implementation should be computationally efficient (Bhatt et al., 2020) and enable integration with any phishing detector's API. The approaches surveyed in section 2 individually do not fulfill all these requirements. They either produce explanations that consist of individual words and are therefore difficult to comprehend for lay users (cf. Burdisso et al., 2019), or are tied to highly specific text classification models. However, the *rationales* by Lei et al. (2016) closely resemble the desired explanations, while the algorithm for *anchors* by Ribeiro et al. (2018) is model-agnostic and efficient. In the following, we integrate these two approaches as *document anchors*.

We begin with a text document (e.g., an e-mail) that is represented as a vector of tokens $x = [t_0, t_1, \ldots, t_N]$. A model m (e.g., a phishing detector) takes x as the input and classifies it into class c ($m(x) = c$). The goal is to find an explanation for the classification of x into c. As outlined above, the explanation should convey which tokens t_i or sequences of tokens were decisive for the classification. An anchor candidate is represented as a binary vector a of the same length as x. $a_i = 1$ ($a_i = 0$) indicates that the token at the position i remains (is replaced). In analogy to the notation used in Ribeiro et al. (2018), we define

$$A_a(x) = \begin{cases} 1 & \text{if } x \odot a = a \\ 0 & \text{otherwise} \end{cases}. \quad (1)$$

Then, a *document anchor* is an A_a for which

$$A_a(x) = 1 \text{ and } \mathbb{E}_{\mathcal{D}_x(z|A_a)}\left[m(x) = m(z)\right] \geq 1 - \tau, \quad (2)$$

where $\tau \approx 0$ is a constant that defines the confidence bound. The perturbation set $\mathcal{D}_x(z)$ is the (potentially infinite) set of vectors z that are generated by replacing a fraction of tokens in x with empty, unknown (out of vocabulary), or similar tokens (e.g., as determined by a language model) (Ribeiro et al., 2018; Fernandez et al., 2019). $\mathcal{D}_x(z|A_a) = \{z \in \mathcal{D}_x(z) | A_a(z) = 1\}$ is the subset of $\mathcal{D}_x(z)$ to which A_a applies.

3.1. Objective

In general, many anchors exist for any given x, but not all of them constitute a good explanation (cf. Ribeiro et al., 2018). For example, A_{a_1} where $a_i = 1 \forall i$ is always an anchor, but conveys no information to the user that is particularly helpful in distinguishing between phishing and genuine e-mails.

Therefore, the search for an anchor is guided by an objective function $\mathcal{O}(A_a)$ that encodes the desired properties.

Similar to the *rationales* of Lei et al. (2016), document anchors should be short sequences of tokens. However, the shortest possible anchor is not necessarily the best explanation in the eyes of the user. A few distinctive words might be sufficient to guarantee the correct classification, but the user perceives text in phrases (Burdisso et al., 2019). Hence, we introduce a *Length* measure

$$\text{Length}_l(a) = (|a| - l)^2 , \tag{3}$$

which allows to specify a target length l. To ensure that sequences rather than individual tokens are selected, *Coherence* measures how many separate sequences of tokens are selected (Lei et al., 2016):

$$\text{Coherence}(A_a) = \sum_i |a_i - a_{i-1}|. \tag{4}$$

If required, other proxy measures can be constructed and utilized. The objective function $\mathcal{O}(A_a)$ is a linear combination of proxy measures. Its coefficients weight the different proxy measures against each other. The design of the objective function is the primary means by which the first requirement, the comprehensibility of explanations, is ensured in a given use case. By convention, $\mathcal{O}(A_a)$ is constructed such that a minimal value represents an optimal anchor.

3.2. Search Algorithm and Strategies

We use a beam search algorithm to find a document anchor A_a that minimizes the objective function $\mathcal{O}(A_a)$ (cf. Ribeiro et al., 2018). We begin with N seed candidates. In each step, $N_{\text{child}} < N$ child candidates are generated from each candidate by evolving it according to a search strategy. We then use the KL-LUCB algorithm (Kaufmann & Kalyanakrishnan, 2013) to determine the N best candidates among the children. To this end, we obtain an estimate of the expectation value

$$\mathbb{E}_{\mathcal{D}_x(z|A_a)} \left[m(z) = m(x) \right]. \tag{5}$$

by computing the model's prediction given a $z \in \mathcal{D}_x(z)$ for which $A_a(z) = 1$. Note that in line with the specified requirements, the algorithm makes no assumptions regarding the model's inner workings. We repeat this computation until the lower bound on the expectation value of the N^{th}-best candidate surpasses the upper bound on the next-best candidate's expectation value by at least Δ_{min}.

The search strategy, i.e., the method by which children are generated from previous candidates, strongly affects how quickly an optimal anchor is found and thus the algorithm's efficiency. Further, in combination with the beam search parameters N and N_{child}, the choice of a search strategy influences the consistency of the anchors' properties. Ribeiro et al. (2018) build their *anchors* from the ground up, creating candidates by adding one word at a time. This is unsuitable in the case of token-based explanations, which aim to uncover phrases (cf. Burdisso et al., 2019; Lei et al., 2016). Instead, new candidates are generated by growing, shrinking, or shifting the highlighted sequences of tokens, erasing them or seeding new ones. In the case of long documents, an anchor candidate can be split into multiple parts, each of which is evolved according to a different strategy.

3.3. Implementation

As organizations receive large quantities of e-mails per day, efficiency is a major concern both in the design and the implementation of an algorithm to be applied for phishing prevention. To date, XAI algorithms are often developed as tools that researchers and data scientists run on a single machine to investigate and debug models they can access directly (Bhatt et al., 2020). For application at large scales, XAI systems need to be efficient and scalable, as well as configurable and testable (cf. Martin, 2018). To address these requirements, we designed and developed a prototypical implementation[1] of the algorithm described above.

The implementation separates the sampling from $\mathcal{D}_x(z)$ and the search for an anchor A_a into independent components. To ensure that it is truly model-agnostic, the sampling component does not require direct access to the model, but only a prediction API that takes a document z and returns the classification $m(z)$. Hence, it does not depend on any particular machine-learning framework and leaves load balancing and scaling of model instances to the model's serving infrastructure. This further has the operational advantage that the model can be queried with optimal batch size and guarantees that explanations are always generated based on the model currently deployed in production. To not slow the search by potentially high-latency model queries, the samples from $\mathcal{D}_x(z)$ are transferred to the search component through a message queue, which enables back-pressure to build up. If an upper limit on the number of model queries is desirable, the search component can store the samples in a ring buffer. As it handles only the abstract vectors a, the search component's implementation is independent not only of the model, but also the representation of the documents. It can thus be based on highly optimized array structures. Similarly, the bound estimation for the KL-LUCB algorithm can be delegated to a fast machine-level implementation.

4. Application to Phishing Prevention

Distinguishing phishing e-mails from genuine e-mails is a difficult task for users (Parsons et al., 2016), especially in the case of personalized phishing attempts (Williams

[1] https://github.com/kluge-ai/docanchors

Table 1. Typical textual cues in phishing e-mails (Parsons et al., 2016; Kim & Kim, 2013; Williams et al., 2018).

CLASS	EXAMPLES FROM THE DATASET
Urgency	*You have 72 hours to verify the information, ...*
Authority	*A Message From The CEO ...*
Importance	*We have reason to believe that your account was accessed by a third party.*
Reward/Positive Consequence	*In return we will deposit $70 to your account ...*
Loss/Negative Consequence	*If you do not verify yourself, your account will be suspended.*
References to Security and Safety	*Security is one of our top goals at our company, ...*
Spelling and Grammatical Errors	*You were qualified to participate in $50.00 reward surwey.*
Lack of Personalization	*Dear Valued Customer, ...*

et al., 2018; Kim & Kim, 2013). The main discriminatory elements are the sender's address and other technical information in the e-mails' header, the links included in the e-mail, and the text itself (Parsons et al., 2016; Williams et al., 2018). In contrast, the graphical design of the e-mail, visual elements, and the presence of legal information (e.g., a disclaimer) are of little informative value (Parsons et al., 2016). With our document anchors approach, we focus exclusively on textual cues. On the one hand, textual cues are easiest to comprehend and evaluate for lay people (cf. Parsons et al., 2016). On the other hand, they are the only cue present in technically inconspicuous variants of phishing e-mails (cf. Williams et al., 2018). Further, in contrast to technical cues (e.g., URL spoofing), they often cannot be unambiguously detected by automated filters (cf. Pienta et al., 2018). Table 1 summarizes typical classes of textual cues reported in the literature.

4.1. Instantiation

To assess the suitability of document anchors as an anti-phishing measure, we demonstrate and evaluate the algorithm using a real-world phishing dataset. The IWSPA-AP v2.0 dataset (Zeng et al., 2020) consists of 503 phishing and 4082 legitimate e-mails and was compiled to allow for the comparison of machine-learning based phishing detectors. For the following experiments, we randomly select 80% of e-mails as the training set on which we train a small bidirectional LSTM as the phishing detector. The remainder of the e-mails serves as the test set. We calibrate the phishing detector to consider e-mails with ambiguous scores as suspicious. To this end, we take the minimal (maximal) score assigned to phishing (genuine) e-mails in the test set as the lower (upper) threshold. This results in the classification of 119 genuine e-mails (15%) and 8 phishing e-mails (8%) from the test set as suspicious. For our evaluation, we focus on the latter. We code these e-mails (at the level of individual tokens) according to the classes listed in Table 1.

We instantiate the document anchors algorithm to explain the classification of e-mails as suspicious. To generate $z \in \mathcal{D}_x(z)$, we replace tokens in the e-mail with empty tokens. We choose the replacement probability p_r such that $\mathbb{E}\left[m(x) = m(z)\right] = 0.5$, which results in an unbiased estimation of the expectation value in equation (5). To obtain a reasonable probability that an anchor candidate (which contains sequences of tokens) matches z, we always replace connected sequences s of seven tokens. To maximize p_r and minimize the number of model calls, we bias the selection of tokens for replacement based on the phishing detector's score for the respective sequence. As we are interested in uncovering why the detector assigned a high score, we consider $m(x) = m(z)$ fulfilled if $m(z) > m(s)_{\min} + 0.5 \cdot (m(s)_{\max} - m(s)_{\min})$, where $m(s)_{\min}$ ($m(s)_{\max}$) is the minimal (maximal) score assigned to any sequence s. We query the detector with a batch size of 256 samples and generate at most 10^4 samples from $\mathcal{D}_x(z)$. Since the detector is sensitive to small excerpts, $p_r \gg 0.5$, which we account for by relaxing the match condition (eq. 1) to 80% overlap. To obtain short segments of highlighted text, we use $\mathcal{O}(A_a) = \text{Coherence}(A_a) + 0.2 \cdot \text{Length}_7(A_a)$ as the objective function. We initialize the search with $N = 5$ seed candidates, each consisting of one randomly placed sequence of three tokens. We generate $N_{\text{child}} = 3$ children by splitting candidates into five randomly sized parts and evolving each with a strategy randomly chosen (biased by p_s) from the following list: Shift sequences of selected tokens forward/backward by one token ($p_s = 0.53$), add ($p_s = 0.26$) or remove ($p_s = 0.11$) one token from each selected sequence, erase all selections ($p_s = 0.05$) or select a new sequence of three tokens ($p_s = 0.05$).

4.2. Evaluation

To analyze whether the document anchors algorithm succeeds at uncovering relevant textual cues, we conduct a functionally-grounded evaluation as defined in the XAI evaluation framework by Doshi-Velez & Kim (2018). For each of the suspicious e-mails, we run the algorithm 100 times and assess the resulting anchors using two proxies: To measure that the anchor indeed has a strong influence on the classification, we take the score that the phishing detector assigns to the highlighted parts of the e-mail (*Score*). As the proxy for the anchor's relevance for identifying a phishing e-mail, we compute the fraction of tokens in the anchor that were coded as a phishing cue (*Relevance*).

Table 2. Results of the functionally-grounded evaluation of the *document anchors* algorithm. For each suspicious e-mail, its length (number of tokens), the score assigned by the phishing detector, and the fraction of tokens coded as phishing cues are given. Anchors were generated using three different approaches and assessed using the *Score* and *Relevance* proxies as described in the main text. The values given for the proxies are the median of the evaluation of 100 anchors, with the mean absolute deviation from the median shown in brackets. The best values obtained for each e-mail are marked.

E-MAIL			Fraction	DOCUMENT ANCHORS		SEARCH ONLY		RANDOM SELECTION	
	Length	Score	of Cues	Score	Relevance	Score	Relevance	Score	Relevance
A	73	.06	.26	**.14**(**.05**)	**.67**(**.26**)	.08(.05)	.27(.28)	.07(.05)	.29(.14)
B	28	.93	.25	**.64**(**.17**)	**.60**(**.24**)	.35(.15)	.00(.22)	.42(.12)	.20(.13)
C	35	.95	.51	**.46**(**.16**)	**.60**(**.25**)	.26(.12)	.50(.26)	.29(.13)	.50(.15)
D	48	.97	.54	**.34**(**.13**)	**.80**(**.24**)	.19(.12)	.63(.26)	.21(.08)	.57(.15)
E	14	.78	.29	**.66**(**.09**)	.25(.16)	.47(.15)	.33(.18)	.54(.12)	**.33**(**.14**)
F	718	.06	.05	**.10**(**.04**)	.00(.10)	.07(.04)	.00(.03)	.07(.03)	.00(.05)
G	34	.95	.50	**.50**(**.15**)	**.60**(**.24**)	.28(.14)	.52(.23)	.32(.12)	.50(.15)
H	44	.10	.50	**.12**(**.04**)	**.59**(**.20**)	.10(.04)	.55(.25)	.09(.05)	.50(.15)

To benchmark the values obtained for the proxies, we utilize two competing approaches: As the baseline, we create anchors by randomly highlighting $l = 7$ tokens in an e-mail. Further, to assess the effect of the sampling component, we independently run the search component with the same objective function as the full approach.

The results are summarized in Table 2. We find that the anchors generated by the full algorithm are classified as suspicious by the phishing detector. Further, the anchors consist largely of tokens coded as phishing cues. We note that the *Relevance* is not only dependent on the document anchors algorithm, but strongly influenced by which tokens most affect the phishing detector's score, which are not necessarily those identified as phishing cues by the human labelers. For example, an in-depth analysis reveals that lack of personalization does not strongly influence the phishing detector's assessment. Comparison with the benchmark approaches reveals that the phishing detector generally yields higher scores for anchors than for either kind of randomly selected tokens. The anchors also best match the tokens coded as relevant for identifying phishing attempts. Analysis of the two e-mails where the anchors' *Relevance* does not surpass that of the random approaches provides valuable insights. For the extremely short e-mail *E*, the specified target length for the anchors exceeds the number of relevant tokens, while the phishing detector assigns similar scores to all $z \in \mathcal{D}_{\mathrm{E}}(z)$. Thus, no anchors that are meaningful to users can be found, illustrating an inherent limitation of the overall approach. For the long e-mail *F*, the mean *Relevance* is significantly higher for document anchors (.10) than for either of the random approaches (.03 and .05). This indicates that the algorithm did not consistently identify the small fraction of tokens labeled as telltale signs of phishing, but successfully generated relevant anchors in the cases where the beam search encountered a match. Thus, increasing N or seeding with $N_{\mathrm{seed}} \gg N$ candidates is expected to improve consistency for long e-mails.

5. Conclusion and Outlook on Future Research

The high susceptibility of users to increasingly sophisticated and personalized phishing attacks poses a threat to organizations and private individuals alike (Pienta et al., 2018). While current anti-phishing measures successfully address particular objectives (e.g., automatic filtering of e-mails or educating users on telltale signs of phishing), they fail to effectively prevent users from falling for phishing attacks (Williams et al., 2018; Dennis & Minas, 2018). Against this background, we proposed to utilize XAI methods to leverage the power of machine-learning based phishing detectors to design an anti-phishing measure that both effectively raises users' awareness and guides their assessment of suspicious e-mails. As a first step towards this goal, inspired by prior work (Lei et al., 2016; Ribeiro et al., 2018), we introduced the *document anchors* algorithm that identifies elements of a text that significantly contribute to its classification by a black-box classifier. Using a real-world dataset and a deep learning phishing detector, we demonstrated that the algorithm succeeds in uncovering cues and phrases in e-mails that are relevant for discriminating between phishing and genuine e-mails. Further, the algorithm's design and implementation address central requirements for the application of XAI systems at large scales.

The work presented in this paper is part of an ongoing effort to design an XAI system for phishing prevention by integrating research on automated phishing detection, phishing susceptibility, and anti-phishing training. Future work will encompass transfer to state-of-the-art phishing detectors based on modern language models, thorough evaluation of the efficiency both of different search strategies and the algorithm's implementation, as well as human-grounded evaluations (Doshi-Velez & Kim, 2018) based on established concepts for the evaluation of anti-phishing measures.

Acknowledgements

We kindly thank Rakesh M. Verma (University of Houston) for providing us the dataset.

References

Albakry, S. and Vaniea, K. Automatic phishing detection versus user training, Is there a middle ground using XAI? In *CEUR Workshop Proceedings*, volume 2151, 2018. URL http://ceur-ws.org/Vol-2151/Paper_P2.pdf.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J., and Eckersley, P. Explainable Machine Learning in Deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 648–657. Association for Computing Machinery, 2020. doi: 10.1145/3351095.3375624.

Burdisso, S., Errecalde, M., and Montes-y-Gomez, M. t-SS3: a text classifier with dynamic n-grams for early risk detection over text streams. 2019. arXiv:1911.06147.

Dennis, A. and Minas, R. Security on Autopilot: Why Current Security Theories Hijack our Thinking and Lead Us Astray. *The DATABASE for Advances in Information Systems*, 49:15–37, 2018. doi: 10.1145/3210530.3210533.

Doshi-Velez, F. and Kim, B. Considerations for Evaluation and Generalization in Interpretable Machine Learning. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pp. 3–17. Springer, 2018. doi: 10.1007/978-3-319-98131-4_1.

Fernandez, C., Provost, F., and Han, X. Counterfactual Explanations for Data-Driven Decisions. In *Proceedings of the Fortieth International Conference on Information Systems*. Association for Information Systems, 2019. URL https://aisel.aisnet.org/icis2019/data_science/data_science/8/.

Kaufmann, E. and Kalyanakrishnan, S. Information Complexity in Bandit Subset Selection. In *Proceedings of the 26th Annual Conference on Learning Theory*, pp. 228–251, 2013. URL http://proceedings.mlr.press/v30/Kaufmann13.html.

Kim, E. and Kim, J. H. Understanding persuasive elements in phishing e-mails. *Online Information Review*, 37(6): 835–850, 2013. doi: 10.1108/OIR-03-2012-0037.

Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., and Hong, J. Teaching Johnny not to fall for phish. *ACM Transactions on Internet Technology*, 10, 2010. doi: 10.1145/1754393.1754396.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1011.

Martens, D. and Provost, F. Explaining Data-Driven Document Classification. *MIS Quarterly*, 38:73–99, 2014. doi: 10.25300/MISQ/2014/38.1.04.

Martin, R. C. *Clean Architecture: A Craftman's Guide to Software Structure and Design*. Pearson Education, 2018.

Parsons, K., Butavicius, M. A., Pattinson, M. R., Calic, D., McCormac, A., and Jerram, C. Do Users Focus on the Correct Cues to Differentiate Between Phishing and Genuine Emails? In *Australasian Conference on Information Systems (ACIS) 2015 Proceedings*, 2016. arXiv:1605.04717.

Pienta, D., Thatcher, J., and Johnston, A. A Taxonomy of Phishing: Attack Types Spanning Economic, Temporal, Breadth, and Target Boundaries. In *Proceedings of the 13th Pre-ICIS Workshop on Information Security and Privacy*. Association for Information Systems, 2018. URL https://aisel.aisnet.org/wisp2018/19/.

Ribeiro, M., Singh, S., and Guestrin, C. Anchors: High-Precision Model-Agnostic Explanations. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1527–1535. Association for the Advancement of Artificial Intelligence, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982/15850.

Williams, E. J., Hinds, J., and Joinson, A. N. Exploring susceptibility to phishing in the workplace. *International Journal of Human-Computer Studies*, 120:1–13, 2018. doi: 10.1016/j.ijhcs.2018.06.004.

Zeng, V., Baki, S., Aassal, A. E., Verma, R., De Moraes, L. F. T., and Das, A. Diverse Datasets and a Customizable Benchmarking Framework for Phishing. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, pp. 35–41. Association for Computing Machinery, 2020. doi: 10.1145/3375708.3380313.