



A REVIEW OF PHISHING DEFENSES: MACHINE-LEARNING DETECTION, PHISHING-RESISTANT AUTHENTICATION, AND HUMAN-CENTERED SECURITY

¹**Mohammad Naeem Kanyar**

¹Independent Researcher

¹Punjab, India

¹naeem.kanyar@gmail.com

Abstract: Phishing continues to be an effective route to account compromise because it exploits both technical weaknesses and ordinary human expectations. Defenses have expanded from blocklists and hand-written rules to machine-learning classifiers, deep-learning models, email authentication, multi-factor authentication, and cryptographic sign-in methods such as passkeys. These approaches solve different parts of the problem, and treating any one of them as a complete solution leaves important gaps. This review examines the main families of phishing defenses, explains what each can and cannot do, and draws practical lessons from research literature and technical standards. The paper uses a structured narrative review of peer-reviewed studies, established books, and authoritative guidance. It compares detection approaches, discusses dataset and evaluation problems, examines the difference between conventional multi-factor authentication and phishing-resistant authentication, and considers the role of user reporting and incident response. Across the reviewed literature, a consistent lesson is that strong results on a static dataset do not automatically translate into dependable protection in a changing environment. Models must be tested on unseen and later-collected data, false positives must be managed, and identity controls should reduce the value of credentials captured by a fraudulent site. The paper proposes a layered defense model that combines message and web filtering, multi-signal detection, phishing-resistant authentication, least-privilege access, easy reporting, and continuous evaluation. This is a literature review; it does not present a newly trained model, original dataset, or experimental performance claims. Its contribution is a practical synthesis intended to help researchers and organizations evaluate phishing defenses as a coordinated security program.

Index Terms - phishing, cybersecurity, phishing website detection, machine learning, deep learning, passkeys, authentication, social engineering, defense in depth.

I. INTRODUCTION

Phishing is a social-engineering attack in which a person is persuaded to trust a message, website, caller, or request that is controlled by an attacker. The aim may be to collect passwords, obtain payment details, deliver malware, steal business information, or gain an initial foothold in an organization. Email is an important channel, but the same approach appears in text messages, social media, collaboration platforms, QR codes, voice calls, and fraudulent sign-in pages. Phishing is therefore not just a problem of filtering unwanted email; it is a problem of protecting decisions and identities across digital services [1]–[4].

The attack works because its technical and human elements reinforce each other. A fraudulent page may look almost identical to a real service, while a message creates urgency or imitates a familiar colleague. Attackers can change domains, use redirectors, compromise legitimate accounts, or tailor content to a particular person. A blocklist may not yet know about a newly created page, a classifier may encounter a pattern missing from its training data, and a user may ignore a warning when the message appears to be part of a normal task. No single defense can be expected to cover every stage.

Research has proposed a wide range of phishing detection methods, including list-based approaches, visual similarity, heuristics, traditional machine learning, and deep learning [1], [5]–[8]. Separately, identity-security guidance has increasingly emphasized authentication methods that bind the sign-in to the legitimate service, rather than relying on users to identify a fake website [9], [10]. These developments point toward a practical question: how should the available controls be combined, and how should their effectiveness be judged?

This paper is a review paper, not an implementation study and not a claim of a new experimental result. It has four objectives: (1) to organize the main technical approaches to phishing detection; (2) to compare their strengths and recurring limitations; (3) to explain how authentication and organizational controls reduce risk beyond detection; and (4) to propose a layered framework and evaluation checklist that can guide future research and deployment.

A. Review questions

The review is guided by four questions: RQ1—What major approaches are used to detect phishing? RQ2—What data and evaluation practices affect the reliability of published results? RQ3—How do authentication and user-centered controls complement detection? RQ4—What practical design principles follow when these defenses are considered together?

B. Contribution and organization

The contribution is a cross-layer synthesis. Rather than ranking algorithms from figures reported in different datasets, the paper explains why such rankings can be misleading and identifies the conditions under which each defense is useful. Section II explains the review method. Section III outlines phishing behavior and detection families. Section IV compares evaluation practices and datasets. Section V reviews identity, user, and organizational controls. Section VI presents a layered framework, Section VII discusses research gaps, and Section VIII concludes.

II. REVIEW METHODOLOGY

A structured narrative review was selected because the topic spans several kinds of evidence: empirical studies of URL and website classifiers, surveys of phishing research, foundational work on social engineering, authentication standards, and organizational cybersecurity frameworks. A single numerical meta-analysis would not be appropriate without comparable datasets, labels, metrics, and experimental settings.

A. Source identification

The source set was assembled around searches and known bibliographies concerning phishing website detection, phishing email classification, machine-learning and deep-learning approaches, usable security, authentication, passkeys, and cybersecurity risk management. Sources include peer-reviewed journal and conference papers, widely cited research on phishing and usable security, relevant standards, and technical guidance from recognized standards bodies. The references include both foundational publications and more recent reviews so that technical approaches can be considered alongside the conditions under which they are used.

B. Inclusion and synthesis

Sources were considered relevant when they addressed at least one of the review questions and supplied a recognizable method, empirical observation, conceptual framework, or authoritative technical requirement. The analysis grouped evidence into detection, data and evaluation, authentication, user-centered security, and organizational response. For each group, the review compared the underlying signal or mechanism, likely benefit, main failure modes, and operational implications. Findings are described qualitatively because published performance figures are not directly comparable across different datasets and study designs.

C. Limitations of the method

This is a structured narrative review, not a registered systematic literature review. It does not claim that every relevant publication was identified, and it does not report a reproducible database-by-database screening flow or a formal risk-of-bias score. The selection may favor accessible English-language sources and well-established work. The conclusions should therefore be read as a reasoned synthesis, not as an exhaustive census of all phishing research. A future systematic review could improve reproducibility by publishing complete search strings, screening decisions, data-extraction fields, and a protocol before selection.

III. PHISHING AND THE MAIN DETECTION APPROACHES

A. The phishing process

A typical campaign can be understood as a sequence: preparation, delivery, persuasion, interaction, and exploitation. Preparation includes choosing a target, registering infrastructure, or compromising a legitimate service. Delivery brings the lure to a user. Persuasion uses a credible identity or a timely request. Interaction may be a click, a file opening, a QR scan, a credential entry, or an authentication approval. Exploitation may then result in account takeover, unauthorized transactions, malware execution, or further movement through a network. This sequence matters because controls can interrupt the attack at different points; prevention does not have to depend on detecting the initial message alone [2]–[4].

B. Blocklists and reputation

List-based systems compare domains, URLs, IP addresses, certificates, or files with known indicators. Their main advantage is speed and operational simplicity: once an indicator is confirmed and distributed, it can be blocked consistently. Their weakness is that they depend on prior observation and timely sharing. New domains, short-lived pages, compromised legitimate sites, and altered URL paths may not match existing records. List-based controls are therefore valuable as a baseline, but they should not be the only detection layer [5], [6].

C. Heuristic and rule-based detection

Heuristic systems look for suspicious properties, such as unusual URL structure, misleading subdomains, excessive redirects, mismatched branding, unexpected credential forms, or inconsistencies between message claims and sender identity. Rules can be transparent and inexpensive to run. However, rules can become brittle as attackers adapt, and legitimate websites sometimes share the same properties as malicious pages. Rules need regular maintenance and should be combined with other evidence rather than treated as proof by themselves [1], [5].

D. Traditional machine learning

Traditional machine-learning systems learn a decision boundary from labeled examples. Models used in the literature include logistic regression, decision trees, random forests, support vector machines, naïve Bayes, and boosting methods. Inputs may describe URL characters and tokens, domain registration details, page behavior, certificate properties, forms, links, or message text. These models can combine weak signals that would be less useful individually. Their quality, however, depends on the labels, feature design, data coverage, and similarity between training conditions and the environment where the model will operate [1], [7], [8].

A model that performs well on one dataset may rely on shortcuts that do not persist. For example, it may learn that particular hosting providers or URL patterns are associated with phishing in the training sample. If attackers move to different

infrastructure, the signal can weaken. Models can also generate false alarms on legitimate pages that contain unusual but harmless features. For these reasons, model selection should consider precision, recall, false-positive burden, calibration, latency, and maintainability—not accuracy alone [7], [8], [11].

E. Deep learning and representation learning

Deep-learning approaches use neural networks to learn patterns from character sequences, words, page content, screenshots, or combinations of inputs. Convolutional neural networks, recurrent networks, long short-term memory networks, and transformer-based architectures have all been explored in security classification more broadly, while phishing research has examined URL-, email-, and website-based models [7], [8], [12]. The appeal is that models can learn useful representations without requiring every signal to be manually specified.

The trade-off is that deep models may require more labeled data, greater compute, more careful monitoring, and additional work to explain their decisions. A large model does not automatically generalize better than a smaller one. Its performance must be evaluated on data that reflects realistic changes in attack campaigns, legitimate websites, language, device types, and browsing behavior. A result obtained from a random split of a static dataset is not enough to establish resilience against future phishing pages.

F. Visual similarity and page-content analysis

Some phishing pages imitate the appearance of a legitimate brand or login screen. Visual approaches compare screenshots, logos, page layout, or other image features; content-based approaches inspect HTML, form behavior, links, and page text. These methods can help identify impersonation even when the URL itself is not obviously suspicious. They can also be costly, depend on page rendering, or be evaded by changes to layout and content. Visual evidence is strongest when combined with URL, reputation, and behavioral signals [1], [5], [13].

G. Email authentication and sender validation

Email authentication mechanisms such as SPF, DKIM, and DMARC help receiving systems assess whether a message is authorized to use a domain and whether signed message content has been altered. They are important controls against certain forms of domain spoofing, but they do not prove that a message is safe. A malicious message can originate from a compromised legitimate account or an attacker-controlled domain that is correctly authenticated. Email authentication should be viewed as one signal within a broader filtering and response process [14]–[16].

Approach	Useful signals	Main strengths	Recurring limitations
Blocklists / reputation	Known URLs, domains, IPs, files	Fast; easy to distribute confirmed indicators	Weak against first-seen or rapidly changing infrastructure
Heuristics / rules	URL patterns, redirects, page behavior	Transparent and comparatively inexpensive	Rules need maintenance and can be evaded
Traditional ML	Engineered URL, domain, page, or message features	Combines multiple signals and supports risk scoring	Sensitive to dataset bias, leakage, and drift
Deep learning	Character sequences, text, page content, images	Can learn complex representations from data	Data/compute needs; explanation and generalization challenges
Visual similarity	Screenshots, logos, layout, page appearance	Useful for brand impersonation	Rendering cost; pages can change appearance
Email authentication	Domain authorization and message signatures	Reduces some forms of sender spoofing	Does not establish that an authenticated sender is trustworthy

Table I. Qualitative comparison of phishing detection approaches.

IV. DATASETS AND EVALUATION: WHY RESULTS NEED CONTEXT

A. Dataset quality

A phishing detector learns from the examples it is given. If the data is incomplete, outdated, duplicated, or poorly labeled, evaluation results may be misleading. Public feeds can supply known phishing URLs, while legitimate examples may come from lists of popular websites or curated collections. These sources are useful but may not reflect the full range of sites that ordinary users visit. A classifier can appear successful if the phishing and legitimate samples differ in a way that is easy to learn but unrelated to the underlying threat [1], [7], [8].

Dataset age is also important. Phishing campaigns change quickly, and labels may become stale when a site is taken down or repurposed. Researchers should document collection dates, labeling rules, deduplication, feature availability, and the handling of unavailable pages. The use of data from one source for malicious samples and another source for benign samples can create source-specific artifacts. Such artifacts may inflate scores if the model learns where a record came from rather than whether it is malicious.

B. Data leakage and unrealistic splits

A random train-test split can place near-duplicate URLs, pages from the same campaign, or examples from the same domain in both training and test sets. This makes the task easier than the real-world problem of identifying a new campaign. Leakage can also occur if features include information gathered after the detection decision would have been made. Stronger evaluation separates data by time, domain, campaign, or collection source where appropriate, and it explains how those separations were constructed [7], [8], [11].

C. Metrics beyond accuracy

Accuracy is the proportion of all predictions that are correct, but it can hide poor performance when legitimate traffic greatly outnumbers phishing traffic. Precision asks what fraction of flagged examples are truly malicious; recall asks what fraction of malicious examples are detected. F1-score summarizes precision and recall, while false-positive rate measures the share of legitimate examples incorrectly flagged. In a real deployment, false positives can interrupt work and encourage users to bypass controls, whereas false negatives allow attacks through. Thresholds should therefore reflect the costs of each error and the actions taken at each risk level.

A complete evaluation should also consider inference latency, throughput, model size, calibration, analyst workload, and performance over time. If the detector is embedded in a browser or email gateway, even a small delay can affect user experience.

If it sends many legitimate pages to manual review, the organization may not have enough staff to investigate them. A paper that reports only a headline accuracy figure leaves these practical questions unanswered.

D. Reproducibility and external validity

Reproducible work should describe the data sources, time window, preprocessing, feature engineering, model settings, train-validation-test split, and evaluation metrics. Where data cannot be redistributed because of privacy or safety concerns, researchers should provide enough detail for the study to be audited and repeated as far as possible. External validity asks whether the findings are likely to hold beyond the particular dataset, language, organization, or period examined. Temporal and cross-source testing are especially important because attackers continually change infrastructure and lures [7], [8], [11].

Metric or practice	Question answered	Why it matters
Precision	How many flagged items are actually malicious?	Low precision creates alert fatigue and unnecessary disruption.
Recall	How many malicious items are detected?	Low recall leaves more attacks undetected.
False-positive rate	How often is legitimate activity flagged?	False alarms can undermine trust in the system.
Temporal test set	Does the model work on later examples?	Tests whether results survive some real-world change.
Cross-source test	Does performance transfer to another feed or environment?	Helps expose source-specific shortcuts.
Latency and throughput	Can the control operate at the required scale?	A detector must fit the operational setting.
Calibration	Do risk scores correspond to observed risk?	Supports sensible thresholds and escalation policies.

Table II. Evaluation measures and practices recommended for phishing detection research.

V. AUTHENTICATION, USERS, AND ORGANIZATIONAL CONTROLS

A. Passwords and multi-factor authentication

Passwords remain vulnerable to reuse, credential stuffing, deceptive login pages, and theft from compromised devices. Multi-factor authentication (MFA) reduces the risk that a password alone is enough to access an account. However, MFA methods differ. SMS codes, email codes, time-based one-time passwords, and push approvals can be more secure than passwords alone but may still be vulnerable to real-time relay, SIM-related risks, or approval manipulation. The phrase “MFA enabled” therefore does not, by itself, tell an organization how resistant an account is to phishing [9], [10], [17].

B. Phishing-resistant authentication and passkeys

NIST defines phishing resistance around preventing authentication secrets or valid outputs from being disclosed to an impostor verifier without depending on the user's ability to notice the deception [9]. WebAuthn and FIDO-based authentication use public-key cryptography and bind authentication to the relying party's identity, making credentials much harder to relay to a fake site [10], [18], [19]. Passkeys can be stored on a device or synchronized through an ecosystem, depending on the implementation. These mechanisms are a strong option for high-impact accounts, especially administrators, finance staff, remote access, and primary email.

Phishing-resistant authentication is not a complete account-security program. Recovery procedures, device loss, help-desk checks, enrollment, session management, and legacy applications still matter. If an attacker can persuade a help desk to reset an account through weak identity verification, the strength of the normal sign-in method may be bypassed. Organizations should evaluate the full lifecycle of authentication, not only the login ceremony [9], [10], [18].

C. User awareness and usable security

Training can help users recognize suspicious requests, but it cannot be expected to prevent every mistake. Phishing messages can be personalized, timed around work tasks, or sent from compromised accounts. Research on usable security has repeatedly shown that the design of a security mechanism affects whether people understand and follow it [20]–[23]. Warnings should explain the risk in plain language, avoid unnecessary interruptions, and provide a safe alternative action. Reporting should be easy enough that users can act quickly even when they are unsure.

A healthy reporting culture is more valuable than a punitive one. If users fear blame after clicking a suspicious link, they may delay reporting, giving an attacker more time. Organizations should make it clear how to report messages, what information to include, and what happens next. Training exercises should be designed to improve detection and response, not simply to rank or embarrass individuals. Measurements should emphasize reporting behavior, time to triage, and the quality of organizational response as well as click rates [20]–[24].

D. Least privilege and incident response

A successful phishing interaction should not automatically grant broad access. Least privilege, separate administrator accounts, conditional access, session controls, and additional approval for sensitive actions can limit what a compromised identity can do. Monitoring can identify unusual sign-ins, suspicious mailbox rules, new forwarding destinations, unexpected changes to recovery settings, and unusual data access. Response playbooks should include revoking sessions, resetting credentials, checking mailbox and identity changes, blocking confirmed indicators, and assessing whether sensitive data was accessed.

E. Governance and risk management

The NIST Cybersecurity Framework 2.0 groups cybersecurity outcomes into Govern, Identify, Protect, Detect, Respond, and Recover [25]. This is a useful way to position phishing as an organizational risk rather than only a filtering problem. Governance establishes ownership and acceptable risk; identification highlights high-value accounts and workflows; protection deploys controls; detection monitors suspicious activity; response contains incidents; and recovery restores trusted access and improves the process. Privacy should also be considered: organizations should limit collection of personal content, define retention periods, restrict access to security telemetry, and explain monitoring practices.

VI. A LAYERED FRAMEWORK FOR PRACTICAL PHISHING DEFENSE

The reviewed literature suggests that the strongest practical design is a set of complementary controls. The following framework is a synthesis proposed by this paper; it is not a tested product or an experimentally validated model. It is intended as a planning structure that can be adapted to the size, risk, and technical maturity of an organization.

Layer 1: Reduce exposure at the boundary

Use email filtering, URL reputation, browser protection, attachment analysis, and appropriate domain-authentication policies. Confirmed indicators should be shared with relevant controls quickly. Allow-lists should be narrow, documented, and reviewed, because a broad exception can bypass several layers at once.

Layer 2: Combine multiple detection signals

Combine reputation with URL structure, page behavior, sender context, message characteristics, and identity telemetry where lawful and appropriate. Use machine-learning scores as one input to a policy. High-confidence malicious content may be blocked, uncertain cases may be isolated or warned about, and low-risk content may proceed with monitoring. Model versions, thresholds, and changes should be documented.

Layer 3: Make phishing-resistant authentication the goal

Prioritize passkeys or security keys for high-risk identities. Where older applications prevent immediate migration, use the strongest supported MFA method and restrict access through device, location, and risk policies where appropriate. Keep a migration register that records exceptions, owners, and target dates. Recovery and help-desk workflows must be reviewed alongside the primary authenticator.

Layer 4: Limit the damage

Use least privilege, separate administrative identities, short-lived sessions where suitable, and stronger approval for sensitive actions such as changing payment details or exporting large volumes of data. These controls reduce the value of credentials even if an attacker succeeds in obtaining them.

Layer 5: Make reporting and response straightforward

Give users a simple reporting mechanism and a clear path for urgent cases. The response team should triage the report, safely inspect links and message details, block confirmed indicators, revoke affected sessions, secure the account, and check for persistence such as mailbox rules or forwarding. The reporting user should receive an appropriate acknowledgement or follow-up.

Layer 6: Measure, learn, and improve

Review performance regularly and after major incidents. Useful measures include the number of confirmed phishing messages reaching users, false-positive rates, median time from report to triage, containment time, phishing-resistant authentication coverage for high-risk accounts, and model performance on later data. Metrics should be used to identify system weaknesses and allocate resources, not to shame individual employees.

Layer	Primary purpose	Example evidence of operation
Boundary controls	Reduce exposure to known and suspicious content	Filtering outcomes, confirmed indicators blocked
Multi-signal detection	Identify patterns not captured by lists alone	Temporal precision/recall, false-positive rate
Phishing-resistant identity	Reduce credential relay and account takeover	Coverage of high-risk accounts; exception register
Least privilege	Limit post-compromise actions	Privilege reviews; sensitive-action approvals
Reporting and response	Shorten time to containment	Report-to-triage and containment times
Continuous improvement	Detect drift and learn from incidents	Periodic review, model versioning, lessons learned

Table III. Proposed layers and example operational evidence.

VII. DISCUSSION AND RESEARCH GAPS

A. Why no single detector is enough

The approaches differ in what they observe. A blocklist asks whether an indicator is already known; a heuristic asks whether a feature looks suspicious; a machine-learning model estimates risk from learned patterns; a visual detector looks for impersonation; and email authentication checks aspects of domain authorization and message integrity. Because their signals are different, combining them can improve resilience. At the same time, stacking controls without considering their interactions can create complexity, latency, duplicated alerts, and unclear responsibility. A layered design needs a clear decision policy and ownership for each control.

B. Better evaluation of changing threats

One research priority is more realistic evaluation. Future work should report temporal splits, cross-source tests, campaign-level deduplication, and the effect of class imbalance. Results should include precision, recall, false-positive burden, calibration, and latency. A model that performs well on a benchmark but fails when a campaign changes is not necessarily useful in production. Researchers should also explain whether features are available at the moment a decision must be made.

C. Explainability, privacy, and fairness

Phishing detection may involve email content, URLs, screenshots, and user behavior. These data can reveal sensitive information. Future systems should minimize collection, control access, and investigate privacy-preserving approaches where feasible. Explainability is also important for analysts who need to understand why a message was blocked or escalated. Evaluation should consider different languages, accessibility needs, devices, and organizational roles so that a detector does not perform poorly for groups underrepresented in its training data.

D. Authentication lifecycle and recovery

Passkeys and security keys reduce some forms of phishing, but research and deployment practice must pay close attention to recovery, account migration, shared devices, device replacement, and help-desk procedures. A secure login can be undermined by a weak fallback. Studies should evaluate both technical resistance and the user experience of enrollment and recovery, particularly for organizations that cannot replace all legacy systems at once.

E. Human-centered measurement

Click rates are easy to count but do not fully represent security. A user may click and immediately report, while another may avoid clicking but fail to report a dangerous message. Future studies should measure reporting, time to report, response quality, repeated exposure, and the friction caused by controls. The aim is to build an environment in which a mistake is less likely to become a breach and suspicious activity reaches the response team quickly.

F. Practical implications

Small organizations may not have the resources to train custom models or operate a dedicated security operations center. They can still apply the framework by prioritizing strong identity protection, managed email and browser security, least privilege, straightforward reporting, and an incident-response checklist. Larger organizations may add in-house detection, threat intelligence, and model monitoring. In both cases, the best control mix depends on the value of the assets, the channels used, and the organization's capacity to maintain the controls.

VIII. CONCLUSION

This review examined phishing defense as a combination of detection, identity protection, user support, and organizational response. Blocklists and reputation controls are efficient against known indicators; heuristics provide transparent signals; machine-learning and deep-learning methods can combine complex patterns but depend on sound data, careful evaluation, and ongoing monitoring. Visual analysis and email authentication contribute useful evidence, but neither establishes that every message or page is safe.

Authentication is a critical complementary layer. Conventional MFA reduces some risks from stolen passwords, but manually entered codes and approval prompts can remain vulnerable to real-time relay or manipulation. Phishing-resistant authentication based on cryptographic origin binding offers stronger protection, provided that enrollment, recovery, and session management are also secure. Least privilege and well-practiced incident response reduce the consequences when prevention fails.

The paper proposes a six-layer framework covering boundary controls, multi-signal detection, phishing-resistant identity, least privilege, reporting and response, and continuous improvement. Its main recommendation is to evaluate these measures as one program rather than as isolated products. Because this is a narrative review, the framework remains a synthesis for future testing. Further empirical work should validate it in real environments and report security outcomes alongside usability, cost, privacy, and performance over time.

REFERENCES

- [1] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University – Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, 2023, doi: 10.1016/j.jksuci.2023.01.004.
- [2] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: A systematic literature review," *Knowledge and Information Systems*, vol. 64, pp. 1457–1500, 2022, doi: 10.1007/s10115-022-01672-x.
- [3] M. Jakobsson and S. Myers, Eds., *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley, 2006.
- [4] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2006, pp. 581–590, doi: 10.1145/1124772.1124861.
- [5] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA: A content-based approach to detecting phishing web sites," in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, 2007, pp. 639–648, doi: 10.1145/1242572.1242659.
- [6] A. Y. Fu, L. Wenying, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on Earth Mover's Distance (EMD)," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 301–311, 2006, doi: 10.1109/TDSC.2006.50.
- [7] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 2015–2028, 2019, doi: 10.1007/s12652-018-0798-z.
- [8] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or not phishing? A survey on the detection of phishing websites," *IEEE Access*, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.
- [9] D. Temoshok et al., *Digital Identity Guidelines: Authentication and Authenticator Management*, NIST SP 800-63B-4, National Institute of Standards and Technology, 2025, doi: 10.6028/NIST.SP.800-63b-4.
- [10] FIDO Alliance, "Passkeys," FIDO Alliance technical information, <https://fidoalliance.org/passkeys/> (accessed October 2026).
- [11] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 99–108, doi: 10.1145/1014052.1014066.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [13] L. Wenying, G. Huang, X. Liu, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," in *Proceedings of the 14th International World Wide Web Conference (WWW 2005)*, pp. 1060–1061, 2005, doi: 10.1145/1062745.1062868.
- [14] S. Kitterman, *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email*, RFC 7208, Internet Engineering Task Force, 2014, doi: 10.17487/RFC7208.
- [15] D. Crocker, T. Hansen, and M. Kucherawy, *DomainKeys Identified Mail (DKIM) Signatures*, RFC 6376, Internet Engineering Task Force, 2011, doi: 10.17487/RFC6376.
- [16] M. Kucherawy and E. Zwicky, *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*, RFC 7489, Internet Engineering Task Force, 2015, doi: 10.17487/RFC7489.
- [17] Internet Engineering Task Force, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 9846, 2026. Available: <https://www.rfc-editor.org/rfc/rfc9846.html>.
- [18] World Wide Web Consortium (W3C), *Web Authentication: An API for Accessing Public Key Credentials—Level 2*, W3C Recommendation, 2021, <https://www.w3.org/TR/webauthn-2/>.

- [19] FIDO Alliance, Client to Authenticator Protocol (CTAP), FIDO2 specifications, <https://fidoalliance.org/specifications/> (accessed October 2026).
- [20] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A systematic review on deep-learning-based phishing email detection," *Electronics*, vol. 12, no. 21, Art. no. 4545, 2023, doi: 10.3390/electronics12214545.
- [21] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 373–382, doi: 10.1145/1753326.1753383.
- [22] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, "Protecting people from phishing: The design and evaluation of an embedded training email system," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2007, pp. 905–914, doi: 10.1145/1240624.1240760.
- [23] J. Downs, M. Holbrook, and L. F. Cranor, "Decision strategies and susceptibility to phishing," in *Proceedings of the Second Symposium on Usable Privacy and Security (SOUPS)*, 2006, pp. 79–90, doi: 10.1145/1143120.1143131.
- [24] M. A. Sasse, S. Brostoff, and D. Weirich, "Transforming the 'weakest link'—a human/computer interaction approach to usable and effective security," *BT Technology Journal*, vol. 19, pp. 122–131, 2001.
- [25] C. Pascoe, S. Quinn, and K. Scarfone, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST CSWP 29, National Institute of Standards and Technology, 2024, doi: 10.6028/NIST.CSWP.29.
- [26] P. A. Grassi, M. E. Garcia, and J. L. Fenton, *Digital Identity Guidelines: Authentication and Lifecycle Management*, NIST SP 800-63B, National Institute of Standards and Technology, 2017, doi: 10.6028/NIST.SP.800-63b.
- [27] A. Oest, Y. Safaei, A. Doupé, G.-J. Ahn, B. Wardman, and K. Tyers, "PhishFarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pp. 1344–1361, 2019, doi: 10.1109/SP.2019.00049.
- [28] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014, doi: 10.1109/TNSM.2014.2377295.
- [29] A. Bergholz, J. De Beer, S. Glahn, M.-F. Moens, G. Paaß, and S. Strobel, "New filtering approaches for phishing email," *Journal of Computer Security*, vol. 18, no. 1, pp. 7–35, 2010, doi: 10.3233/JCS-2010-0371.
- [30] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, 2007, pp. 60–69, doi: 10.1145/1299015.1299021.
- [31] A. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th International World Wide Web Conference (WWW)*, 2007, pp. 649–656, doi: 10.1145/1242572.1242660.
- [32] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proceedings of the ACM Workshop on Recurring Malcode*, 2007, pp. 1–8, doi: 10.1145/1314389.1314391.
- [33] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, Art. no. 5421046, 2017, doi: 10.1155/2017/5421046.
- [34] National Institute of Standards and Technology, *Guide for Conducting Risk Assessments*, NIST SP 800-30 Rev. 1, 2012, doi: 10.6028/NIST.SP.800-30r1.
- [35] National Institute of Standards and Technology, *Security and Privacy Controls for Information Systems and Organizations*, NIST SP 800-53 Rev. 5, 2020, doi: 10.6028/NIST.SP.800-53r5.