

Development and Implementation of the Mail Security Guardian (MSG) System for Multi-Layer Proactive Email Protection Against Spam, Phishing and Malware

Kolchin Rustam¹ 

¹ SoftLine PJSC, Almaty, Kazakhstan

ABSTRACT: Electronic mail remains the primary delivery mechanism for spam, phishing, business email compromise (BEC), and malware in enterprise environments. Existing Secure Email Gateways (SEGs) based on signature filtering and post-delivery analysis achieve approximately 0.91 F1-score under production conditions – a performance ceiling imposed by their reactive architecture. This paper presents the Mail Security Guardian (MSG), a distributed microservice-based platform implementing pre-delivery multi-layer email protection. MSG integrates five sequential analytical layers: pre-SMTP reputation filtering, protocol authentication (SPF, DKIM, DMARC), content and machine learning analysis, dynamic sandbox execution, and an evidence-aggregating decision engine. Evaluated on a corpus of 3.6 million messages collected over six months from a corporate infrastructure of approximately 2,500 users, the MSG machine learning pipeline achieved Precision of 0.992, Recall of 0.978, F1-score of 0.985, and AUC-ROC of 0.996 – an 7.5 percentage-point improvement over the conventional SEG baseline. Binary confusion matrix analysis confirmed legitimate email detection accuracy of 98.6% and malicious email detection accuracy of 99.3%. Production deployment produced reductions of 89.4% in spam prevalence (18.0% to 1.9%), 92.9% in phishing prevalence (4.2% to 0.3%), and 97.6% in malware prevalence (2.1% to 0.05%). End-to-end analysis latency ranges from 120 to 350 milliseconds, with per-node throughput of 50,000 messages per hour supporting linear horizontal scaling. Future directions include adversarial hardening against large language model (LLM)-generated phishing and federated learning for cross-deployment model updates.

KEYWORDS: NDVI, email security; phishing detection; spam filtering; machine learning; sandbox analysis; zero trust; pre-delivery protection; multi-layer defense.

1. Introduction

Electronic mail constitutes the dominant corporate communication channel and the primary attack surface for modern cyber threats. The Anti-Phishing Working Group documented 4,987,809 phishing attacks in 2023, surpassing 4.7 million in the preceding year (Anti-Phishing Working Group, 2023). Machine learning research confirms that bulk spam, credential-harvesting phishing, and malware delivery operate through the same channel yet require fundamentally different detection logics (Ahmed et al., 2022). The convergence of three distinct attack categories within a single traffic stream creates a combinatorial detection challenge that single-modality security controls cannot adequately

address.

The taxonomy of email-borne threats spans four principal categories. Unsolicited bulk email – spam – imposes direct productivity costs and distributes downstream phishing and malware. Spear phishing and Business Email Compromise (BEC) represent targeted deception attacks that exploit communicational trust relationships resistant to conventional content filtering (Atlam & Oluwatimilehin, 2023). Malware delivery embeds executable content in attachments or hyperlinks directing recipients toward drive-by download infrastructure. Each category demands distinct detection logic, demanding integrated simultaneous analysis rather than sequential category-specific filtering.

Traditional Secure Email Gateways (SEGs) apply signature matching and reputation lookups at or after message delivery. This architecture embeds a fundamental limitation: threats can only be detected after corresponding signatures or reputation entries exist. Zero-day malware variants, newly registered phishing domains, and socially engineered BEC messages traverse signature-based controls unimpeded. Research on spammer strategies documents systematic exploitation of signature update latencies (Jáñez-Martino et al., 2022). Post-delivery remediation – retrieving messages from recipient mailboxes – is operationally complex and frequently infeasible after messages have been read.

Machine learning methods achieve superior detection rates compared to static rule-based approaches. Ensemble architectures combining gradient-boosted decision trees with neural language models capture both structured feature signals and semantic content properties (Innab et al., 2024; Kumar et al., 2024). Transformer-based language models demonstrate additional discriminative capacity for spear phishing messages that closely mimic legitimate organizational writing (Jamal et al., 2024; Otieno et al., 2023). Sandbox-based dynamic analysis provides complementary behavioral evidence: executing suspicious files in isolated environments and observing runtime API calls, network connections, and filesystem modifications reveals malicious intent independent of signature coverage (Aboaoja et al., 2023). Email authentication standards SPF, DKIM, and DMARC establish a cryptographic provenance chain enabling detection of domain spoofing (Ragheb et al., 2023).

The research gap lies in integrating these complementary mechanisms into a coherent multi-layer architecture applying all analytical modalities before message delivery – a pre-delivery model – rather than relying on any single method or applying analysis post-delivery.

This paper presents MSG, a distributed microservice-based email security platform implementing pre-delivery multi-layer analysis. MSG was evaluated over six months on 3.6 million messages from a 2,500-user infrastructure. Principal contributions are: (1) a reference architecture for pre-delivery multi-layer email security integrating five analytical layers within a horizontally scalable microservice model; (2) a validated machine

learning pipeline achieving F1-score of 0.985 and AUC-ROC of 0.996, improving 7.5 points over the conventional SEG baseline; (3) production deployment results demonstrating 89.4%, 92.9%, and 97.6% reductions in spam, phishing, and malware prevalence.

2. Related Work

2.1 Machine Learning for Email Threat Detection

Ensemble tree-based methods achieve strong results on structured feature representations combining URL lexical properties, header patterns, and sender metadata. Innab et al. (2024) demonstrated stacked ensemble classifiers outperform individual baseline classifiers on phishing detection benchmarks, reducing false negative rates for zero-day phishing variants. Adnan et al. (2024) validated the stacking ensemble approach for spam classification, confirming that heterogeneous base learner combinations provide superior accuracy while maintaining real-time inference latency. XGBoost delivers consistently competitive performance compared to more computationally expensive deep learning alternatives when operating on structured URL and domain features (Almujahid et al., 2024; Kumar et al., 2024).

Transformer-based language models extend the representational capacity beyond manually engineered feature spaces. Jamal et al. (2024) demonstrated fine-tuned DistilBERT and RoBERTa models achieve test accuracy up to 99% on balanced email classification datasets, with performance gains concentrated on messages exploiting contextual coherence to evade lexical pattern matching. Otieno et al. (2023) confirmed BERT transfer learning effectiveness for phishing classification, while Altwaijry et al. (2024) found transformer architectures outperform convolutional and recurrent baselines across multiple evaluation metrics when trained on combined content and header metadata. Doshi et al. (2023) reported that a dual-layer architecture combining transformer content classification with URL metadata analysis produces measurably lower false negative rates than content-only classification on a combined phishing-and-spam evaluation set.

URL lexical features provide domain-agnostic phishing signals generalizing across newly registered domains absent from training-time blocklists. Salloum et al. (2022) identified domain character entropy, subdomain depth,

and Levenshtein edit distance to brand domain watchlists as consistently high-importance predictors. Alnemari and Alshammari (2023) demonstrated that Levenshtein distance functions as a particularly discriminative typosquatting detector, identifying domains differing from legitimate organizational identities by one- or two-character substitutions.

2.2 Sandbox-Based Dynamic Malware Analysis

Static analysis fails against packed or obfuscated malware modifying structural representation to evade signatures. Liu et al. (2022) demonstrated that sandboxes augmented with emulated user behavioral signals – synthetic mouse movement and keyboard interaction – significantly improve detection rates for malware samples that suppress behavioral indicators until confirming execution in a human-operated environment. Aboaoja et al. (2023) specifically addressed evasive malware that delays initial behavioral indicator manifestation beyond the observation window of time-limited sandboxes, showing that extended dynamic extraction improves detection of samples that evade standard static analysis tools. Azeem et al. (2023) confirmed that dynamic behavioral features – API call sequences, network activity, filesystem modification patterns – provide substantially higher classification accuracy than static structural features for novel malware variants absent from signature databases.

Sandbox evasion techniques – virtualization artifact detection, timing-based delays, human interaction triggers – represent an active adversarial dimension. Afianian et al. (2019) catalogued these evasion mechanisms in a comprehensive survey, establishing the taxonomy that informs defensive sandbox configuration decisions in MSG.

2.3 Email Authentication and Zero Trust

SPF, DKIM, and DMARC constitute a three-layer authentication stack verifying sender identity and providing domain-level policy enforcement. Ragheb et al. (2023) empirically confirmed that SPF and DKIM together provide statistically significant reductions in spoofed message delivery compared to either mechanism independently. Hureau et al. (2024) identified forwarding-induced SPF alignment failures and shared infrastructure configuration complexity as primary obstacles to broader DMARC deployment at scale, while finding that DMARC rejection policies

eliminate direct domain spoofing but leave lookalike domain attacks unaddressed – motivating URL domain similarity analysis in MSG.

Zero Trust architecture eliminates implicit network trust, requiring continuous per-request verification of identity and security posture (Syed et al., 2022). Kang et al. (2023) reviewed Zero Trust application across enterprise security domains, identifying per-session risk assessment as the central mechanism enabling breach containment. Gambo and Almulhem (2025) conducted a systematic literature review finding that Zero Trust adoption reduces breach dwell time and lateral movement propagation. Applied to email, Zero Trust operationalizes as per-message multi-layer evidence aggregation: no message receives delivery authorization based on sender reputation alone.

2.4 Adversarial Robustness and LLM-Generated Phishing

Machine learning email classifiers exhibit degraded performance under adversarially crafted inputs. Gholampour and Verma (2023) demonstrated that fine-tuned ALBERT classifiers achieving 95% clean accuracy suffer 50–70% accuracy reduction under TextAttack adversarial perturbations. Han and Zhang (2022) surveyed adversarial text attack and defense strategies, identifying adversarial training – augmenting the training corpus with algorithmically generated examples – as the most robust defense, with modest clean-accuracy reduction as the primary cost.

The emergence of capable large language models represents a qualitatively distinct escalation in phishing sophistication. Hao et al. (2025) found that at least 51% of observed spam messages and 14% of BEC attacks in their April 2025 dataset exhibited LLM generation characteristics, with LLM adoption in malicious email campaigns accelerating sharply following commercial LLM availability. Roy et al. (2024) confirmed in a controlled red-team setting that LLM-generated phishing achieves substantially higher credibility scores and click-through rates than template-based alternatives. Hazell (2023) found LLM-assisted personalization dramatically reduces manual effort for producing high-quality spear phishing at scale.

3. System Architecture

3.1 Architectural Overview

MSG implements a pre-delivery security model: all analytical processing completes within the MSG infrastructure before any message reaches the destination mail server. MSG accepts inbound SMTP connections on behalf of the protected domain, performs sequential multi-layer analysis, and forwards only passing messages to the internal mail server, holding or rejecting all others.

The system is implemented as a distributed microservice architecture with each analytical layer encapsulated in independently deployable components communicating through Apache Kafka message queues. Containerization through Docker with Kubernetes orchestration manages service lifecycle, autoscaling, and fault recovery. The deployment topology positions the MSG cluster between the public internet and the organization's internal mail infrastructure, processing all inbound traffic through five sequential layers: Pre-SMTP Filtering, Protocol Authentication, Content and ML Analysis, Sandbox Dynamic Analysis, and the Decision Engine.

3.2 Layer 1: Pre-SMTP Filtering

Pre-SMTP filtering operates at the TCP connection and SMTP envelope levels before message content transmission. DNSBL and RBL queries against multiple reputation feeds produce an aggregate sender reputation signal. Greylisting rejects first-encounter connections temporarily, eliminating spam bots that omit RFC-compliant retry behavior while passing legitimate mail transfer agents. Geographic IP filtering applies jurisdiction-level restrictions. Rate limiting suppresses burst-mode delivery from compromised hosts. A formal trust probability $P(\text{trusted} \mid \text{connection metadata})$ governs connection acceptance. Connections below the threshold receive TCP-level rejection, precluding content transmission.

3.3 Layer 2: Protocol Authentication

MSG implements complete SPF (RFC 7208), DKIM (RFC 6376), and DMARC (RFC 7489) verification. SPF verification compares the delivering SMTP source IP against the envelope sender domain's authorized sender list retrieved from DNS. DKIM verification retrieves the cryptographic signature header, obtains the

corresponding public key from DNS, and verifies signature integrity over specified message headers and body. DMARC enforcement applies the sending domain's specified policy – none, quarantine, or reject – while incorporating authentication outcomes as evidence features for subsequent ML classification regardless of policy level.

3.4 Layer 3: Content and Machine Learning Analysis

Text representation applies TF-IDF vectorization to HTML-stripped, Unicode-normalized message body content. A Bayesian classifier provides a computationally efficient first-pass spam probability estimate. Phishing detection applies URL lexical analysis to all extracted hyperlinks: domain character entropy, subdomain depth, hyphen count, presence of IP address literals, and Levenshtein edit distance to the brand domain watchlist, identifying typosquatting domains differing from legitimate identities by one- or two-character operations.

The core classifier applies an ensemble of four architectures: Logistic Regression (linear baseline), Random Forest (bootstrapped trees), XGBoost (regularized gradient boosting), and an MLP neural network (non-linear interactions). Scikit-learn provides the first three implementations; TensorFlow supports MLP training. The ensemble combines complementary representational strengths of linear, tree-based, and neural families. Attachment metadata – MIME type, file extension, archive depth – provides supplementary risk signals.

3.5 Layer 4: Sandbox Dynamic Analysis

Attachment-bearing messages with ML scores exceeding the risk threshold are routed to the sandbox cluster. Static pre-analysis applies PE structure parsing, YARA rule matching against continuously updated pattern libraries, and hash matching against threat intelligence databases, providing immediate classification for known-malicious samples. Dynamic analysis executes samples in an instrumented environment, capturing system API call sequences, network traffic generated during execution, and filesystem and registry modification records. The behavioral trace is scored against risk models trained on annotated behavioral datasets. Dynamic analysis is bounded at 5 seconds per sample. The Kubernetes-orchestrated cluster supports horizontal replica scaling independent of other pipeline

stages.

3.6 Layer 5: Decision Engine

The Decision Engine aggregates layer evidence through a weighted linear scoring function:

$$\text{Score} = \alpha_1 R + \alpha_2 C + \alpha_3 ML + \alpha_4 S$$

where R , C , ML , and S denote the reputation, content, machine learning, and sandbox scores respectively. Weights α_1 – α_4 are calibrated on a validation dataset and updated through periodic retraining. Two configurable thresholds – θ_{deliver} and θ_{reject} – define three disposition zones: delivery, quarantine (pending security operations review), and SMTP-level rejection. The implementation uses Python/FastAPI for internal communication, Go for the high-throughput SMTP proxy, PostgreSQL for structured audit records, and Elasticsearch for full-text message analysis indexing.

4. Methodology

4.1 Dataset and Labeling

The evaluation dataset comprises 3.6 million inbound messages collected over six months from a corporate infrastructure of approximately 2,500 users processing 120,000 messages per day. Message labeling combined three independent evidence streams: user-reported verdicts verified by security operations personnel; sandbox analysis outcomes cross-referenced against multi-engine threat intelligence results; and retrospective threat intelligence enrichment enabling reclassification of initially ambiguous messages after corresponding indicators of compromise were published. This combined annotation strategy avoids the systematic bias of single-source labeling – user reports under-represent successfully deceptive phishing, while sandbox labels are limited to attachment-bearing messages.

4.2 Feature Engineering

The feature vector $x = (x_1, x_2, \dots, x_n)$ concatenates five

feature categories. Reputation and network features encode sender infrastructure signals: DNSBL/RBL aggregate reputation score, geographic source classification, connection behavior metrics, and SPF outcome as a categorical variable covering all RFC-specified result states. Authentication features encode SPF, DKIM, and DMARC outcomes including DKIM signature domain alignment. Text features represent TF-IDF-vectorized message body content. Attachment features capture MIME type, file extension, archive nesting depth, and payload size. URL and behavioral features encode URL count, character entropy statistics, IP literal presence in hostnames, minimum Levenshtein distance to the brand domain watchlist, and structural URL depth.

The binary classification decision function takes the form $D(x) = 1$ if $\text{Score}(x) \geq \theta$, else 0, where $D(x) = 1$ denotes the legitimate class and $D(x) = 0$ denotes the malicious class.

4.3 Training Protocol and Evaluation Metrics

The training protocol adopts a temporal holdout design: the evaluation set consists exclusively of messages received after all training examples. This temporal split is methodologically essential because random splitting introduces data leakage in email security evaluation – threat indicators appearing in both partitions allow classifiers to achieve inflated performance through memorization rather than generalization. A temporal split confronts the classifier with threat patterns not observed during training, accurately representing the operational challenge of detecting novel attack variants.

Class imbalance – legitimate messages substantially outnumber malicious messages in production traffic – is addressed through SMOTE applied to the training partition combined with cost-sensitive weighting during optimization. Classification performance is quantified through Precision ($P = TP/(TP+FP)$), Recall ($R = TP/(TP+FN)$), F1-score ($F1 = 2 \times P \times R / (P+R)$), and AUC-ROC, where ROC plots $TPR = TP/(TP+FN)$ against $FPR = FP/(FP+TN)$ across all decision thresholds.

5. Experimental Results

5.1 Classifier Performance

The MSG machine learning pipeline achieves the following performance on the temporal holdout set:

Metric	Value
Precision	0.992
Recall	0.978
F1-Score	0.985
AUC-ROC	0.996

Precision of 0.992 yields a false positive rate of 0.8%, operationally manageable through quarantine review without disproportionate burden on security personnel. Recall of 0.978 means 97.8% of all malicious messages are

correctly intercepted. AUC-ROC of 0.996 indicates near-perfect class separation across the full threshold range, enabling operational practitioners to adjust the decision boundary to match organizational risk tolerance.

5.2 Binary Confusion Matrix and Baseline Comparison

The binary confusion matrix at the production threshold:

Actual: Legitimate	98.6%	1.4%
Actual: Malicious	0.7%	99.3%

MSG's F1-score of 0.985 is compared against a conventional SEG baseline evaluated under identical conditions on the same temporal holdout set:

System	F1-Score	Improvement
Conventional SEG (baseline)	0.910	–
MSG (proposed)	0.985	+7.5%

The 7.5 percentage-point improvement corresponds to approximately 9,000 additional correctly classified messages per day on the production traffic volume of 120,000 messages per day.

5.3 Scalability

Per-node throughput of 50,000 messages per hour scales linearly with node count:

$$\text{Throughput}_{\text{total}} = n \times T_{\text{node}}$$

End-to-end latency ranges from 120 to 350 milliseconds

for messages processed without sandbox engagement. Sandbox-routed messages incur additional analysis time bounded at 5 seconds per sample. The microservice architecture decouples scaling of compute-intensive components – ML inference and sandbox cluster – from lower-overhead SMTP proxy and authentication services.

6. Deployment Results

MSG was deployed in production at a corporate infrastructure of approximately 2,500 users replacing the existing conventional SEG. Both the pre-deployment reference period and post-deployment measurement period span three months.

Threat Category	Pre-Deployment	Post-Deployment	Relative Reduction
Spam	18.0%	1.9%	-89.4%
Phishing	4.2%	0.3%	-92.9%
Malware	2.1%	0.05%	-97.6%

Spam reduction (89.4%) reflects the combined contribution of pre-SMTP reputation filtering – which blocks bulk delivery from known-malicious infrastructure at the connection level – and ML content classification extending coverage to campaigns absent from DNSBL/RBL feeds at delivery time. Phishing reduction (92.9%) demonstrates multi-layer complementarity: pre-SMTP filtering addresses known infrastructure, authentication verification flags domain spoofing, URL analysis identifies typosquatted domains, and ML classification captures content-level signals not addressed by earlier layers. Malware reduction (97.6%) reflects sandbox effectiveness against both known samples – classified through hash and YARA matching during static pre-analysis – and novel variants identified by behavioral dynamic analysis.

7. Discussion

7.1 Effectiveness and Comparison with Prior Work

The pre-delivery architecture eliminates the fundamental vulnerability of reactive post-delivery models: once a phishing message reaches a user's mailbox, the attack surface includes human decision-making – not subject to software patch updates. The 92.9% phishing reduction observed in production deployment reflects this architectural advantage.

The MSG F1-score of 0.985 compares favorably with reported ranges in the literature. Altwaijry et al. (2024) reported F1-scores of 0.93–0.98 across deep learning models for phishing email detection. Jamal et al. (2024) achieved F1-score of 0.985 with RoBERTa on balanced three-class email datasets. The MSG result at the same threshold reflects the benefit of five-layer evidence integration, which provides the ML classifier with pre-processed reputation and authentication features

augmenting the content-only input space available to standalone text classifiers. Multi-modal feature integration – combining metadata with content – consistently outperforms single-modality approaches (Doshi et al., 2023; Innab et al., 2024).

7.2 Limitations

The ML classifier's dependence on training data representativeness introduces sensitivity to novel threat patterns. Attack techniques absent from the training corpus may not achieve the same detection accuracy as represented patterns – the dataset shift problem identified as a primary challenge in spam detection research (Jáñez-Martino et al., 2022). This motivates a continuous retraining schedule integrating newly labeled examples on a regular cadence. The sandbox computational overhead scales with attachment volume, creating capacity constraints addressable through horizontal cluster scaling. The current implementation does not analyze outbound behavioral patterns for compromised internal account detection – a complementary detection surface addressed by separate user and entity behavior analytics platforms.

7.3 Adversarial Robustness and LLM-Generated Phishing

Hao et al. (2025) found that LLM-generated malicious emails already constituted a majority of observed attack traffic by April 2025, with LLM-generated content exhibiting reduced lexical repetition and higher grammatical coherence that degrades TF-IDF vocabulary-based classification features. Roy et al. (2024) and Hazell (2023) confirmed that LLM-generated phishing achieves substantially higher human credibility assessments compared to template-based alternatives. MSG performance against a threat distribution dominated by

LLM-generated content has not been empirically characterized and represents a priority for future evaluation.

Defenses against LLM-generated phishing include statistical feature analysis targeting characteristic properties of generated text – perplexity distributions, sequence repetition metrics – as supplementary content layer signals (Gholampour & Verma, 2023; Han & Zhang, 2022). Behavioral metadata signals invariant to LLM content quality – sender authentication outcomes, URL registration age, infrastructure reputation – may provide additional discrimination. Federated learning for cross-deployment model updates without centralizing raw message data offers a privacy-preserving pathway to expanding training data diversity.

8. Conclusion

This paper presented MSG, a pre-delivery multi-layer email security platform evaluated on 3.6 million messages over six months from a 2,500-user corporate infrastructure. MSG integrates pre-SMTP reputation filtering, protocol authentication analysis, machine learning content classification, dynamic sandbox analysis, and an evidence-aggregating decision engine within a distributed microservice architecture. The machine learning classifier achieved F1-score of 0.985 and AUC-ROC of 0.996, improving 7.5 percentage points over the conventional SEG baseline of 0.91. Production deployment produced reductions of 89.4% in spam, 92.9% in phishing, and 97.6% in malware prevalence. The pre-delivery architecture provides a scalable, horizontally extensible foundation for proactive corporate email security, eliminating the primary limitation of reactive post-delivery protection models. Future work will address LLM-generated phishing adversarial hardening, federated model update mechanisms, and extended detection and response platform integration.

References

1. Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., & Shah, T. (2022). Machine learning techniques for spam detection in email and IoT platforms: Analysis and research challenges. *Security and Communication Networks*, 2022, 1862888. <https://doi.org/10.1155/2022/1862888>
2. Aboaoja, F. A., Zainal, A. B., Ali, A. M., Ghaleb, F. A., Alsolami, F. J., & Rassam, M. A. (2023). Dynamic extraction of initial behavior for evasive malware detection. *Mathematics*, 11(2), 416. <https://doi.org/10.3390/math11020416>
3. Adnan, M., Imam, M. O., Javed, M. F., & Murtza, I. (2024). Improving spam email classification accuracy using ensemble techniques: A stacking approach. *International Journal of Information Security*, 23, 505–517. <https://doi.org/10.1007/s10207-023-00756-1>
4. Afianian, A., Niksefat, S., Sadeghiyan, B., & Baptiste, D. (2019). Malware dynamic analysis evasion techniques: A survey. *ACM Computing Surveys*, 52(6), 126. <https://doi.org/10.1145/3365001>
5. Almujaheed, N. F., Haq, M. A., & Alshehri, M. (2024). Comparative evaluation of machine learning algorithms for phishing site detection. *PeerJ Computer Science*, 10, e2131. <https://doi.org/10.7717/peerj-cs.2131>
6. Alnemari, S., & Alshammari, M. (2023). Detecting phishing domains using machine learning. *Applied Sciences*, 13(8), 4649. <https://doi.org/10.3390/app13084649>
7. Altwaijry, N., Al-Turaiki, I., Alotaibi, R., & Alakeel, F. (2024). Advancing phishing email detection: A comparative study of deep learning models. *Sensors*, 24(7), 2077. <https://doi.org/10.3390/s24072077>
8. Anti-Phishing Working Group. (2023). Phishing activity trends report: 4th quarter 2023. APWG. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf
9. Atlam, H. F., & Oluwatimilehin, O. (2023). Business email compromise phishing detection based on machine learning: A systematic literature review. *Electronics*, 12(1), 42. <https://doi.org/10.3390/electronics12010042>
10. Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S., & Alzahrani, M. (2023). Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. *Heliyon*, 9(1), e23574. <https://doi.org/10.1016/j.heliyon.2023.e23574>
11. Doshi, J., Parmar, K., Sanghavi, R., & Shekoker, N. M. (2023). A comprehensive dual-layer architecture for phishing and spam email detection. *Computers & Security*, 133, 103378.

- <https://doi.org/10.1016/j.cose.2023.103378>
12. Gambo, L., & Almulhem, A. (2025). Zero trust architecture: A systematic literature review. *Journal of Network and Systems Management*. <https://doi.org/10.1007/s10922-025-09998-x>
13. Gholampour, P. M., & Verma, R. M. (2023). Adversarial robustness of phishing email detection models. In *Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics (IWSPA '23)* (pp. 67–76). ACM. <https://doi.org/10.1145/3579987.3586567>
14. Han, X., & Zhang, T. (2022). Text adversarial attacks and defenses: Issues, taxonomy, and perspectives. *Security and Communication Networks*, 2022, 6458488. <https://doi.org/10.1155/2022/6458488>
15. Hao, W., Tran, V., Rideout, V., Wang, Z., Dasbach-Prisk, A., Afifi, M. H., Yang, J., Katz-Bassett, E., Ho, G., & Cidon, A. (2025). Do spammers dream of electric sheep? Characterizing the prevalence of LLM-generated malicious emails. In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25)* (pp. 192–203). ACM. <https://doi.org/10.1145/3730567.3732922>
16. Hazell, J. (2023). Spear phishing with large language models. *arXiv preprint arXiv:2305.06972*. <https://doi.org/10.48550/arXiv.2305.06972>
17. Hureau, O., Bayer, J., Duda, A., & Korczyński, M. (2024). Spoofed emails: An analysis of the issues hindering a larger deployment of DMARC. In *Passive and Active Measurement (PAM 2024), Lecture Notes in Computer Science (Vol. 14537, pp. 232–261)*. Springer. https://doi.org/10.1007/978-3-031-56249-5_10
18. Innab, N., Osman, A. A. F., Ataelfadiel, M. A. M., Abu-Zanona, M., Elzaghmouri, B. M., Zawaideh, F. H., & Alawneh, M. F. (2024). Phishing attacks detection using ensemble machine learning algorithms. *Computers, Materials & Continua*, 80(1), 1325–1345. <https://doi.org/10.32604/cmc.2024.051778>
19. Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2022). A review of spam email detection: Analysis of spammer strategies and the dataset shift problem. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-022-10195-4>
20. Jamal, S., Wimmer, H., & Sarker, I. H. (2024). An improved transformer-based model for detecting phishing, spam, and ham emails: A large language model approach. *Security and Privacy*, 7(3), e402. <https://doi.org/10.1002/spy2.402>
21. Kang, H., Liu, G., Wang, Q., Meng, L., & Liu, J. (2023). Theory and application of zero trust security: A brief survey. *Entropy*, 25(12), 1595. <https://doi.org/10.3390/e25121595>
22. Kumar, P., Antony, K., Banga, D., & Sohal, A. S. (2024). PhishNet: A phishing website detection tool using XGBoost. *arXiv preprint arXiv:2407.04732*. <https://doi.org/10.48550/arXiv.2407.04732>
23. Liu, S., Feng, P., Wang, S., Sun, K., & Cao, J. (2022). Enhancing malware analysis sandboxes with emulated user behavior. *Computers & Security*, 115, 102613. <https://doi.org/10.1016/j.cose.2022.102613>
24. Otieno, D. O., Namin, A. S., & Jones, K. S. (2023). The application of the BERT transformer model for phishing email classification. In *Proceedings of the 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 1303–1308). <https://doi.org/10.1109/COMPSAC57700.2023.00198>
25. Ragheb, M., Elmedany, W., & Sharif, M. (2023). The effectiveness of DKIM and SPF in strengthening email security. In *Proceedings of the 10th International Conference on Future Internet of Things and Cloud (FiCloud 2023)* (pp. 422–426). <https://doi.org/10.1109/FiCloud58648.2023.00068>
26. Roy, S. S., Thota, P., Naragam, K. V., & Nilizadeh, S. (2024). From chatbots to phishbots? Phishing scam generation in commercial large language models. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP)* (pp. 36–54). <https://doi.org/10.1109/SP54263.2024.00182>
27. Salloum, S. A., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, 10, 65703–65727. <https://doi.org/10.1109/ACCESS.2022.3183083>
28. Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z. A., & Doss, R. R. M. (2022). Zero trust architecture (ZTA): A comprehensive survey. *IEEE*

Access, 10, 57143–57179. <https://doi.org/10.1109/ACCESS.2022.3174679>