PANTAO
The International Journal of the Humanities and Social Sciences
Volume 4, Issue 4
ISSN 3028-0877 | doi 10.69651/2022PIJHSS

Impact analysis of mobile-first AI-assisted Phishing Detection across platforms using Cloud APIs

Adealix Jairon N. Maranan*
Technological University of the Philippines
Taguig City, Philippines
Email: adealixjairon.maranan@tup.edu.ph

Kent S. Madrideo
Technological University of the Philippines
Taguig City, Philippines
Email: kent.madrideo@tup.edu.ph

Sharwin John C. Marbella
Technological University of the Philippines
Taguig City, Philippines
Email: sharwinjohn.marbella@tup.edu.ph

Kevin R. Ofracio
Technological University of the Philippines
Taguig City, Philippines
Email: kevin.ofracio@tup.edu.ph

Paul Dominic A. Syparrado
Technological University of the Philippines
Taguig City, Philippines
Email: pauldominic.syparrado@tup.edu.ph

Pops V. Madriaga
Technological University of the Philippines
Taguig City, Philippines
Email: pops_madriaga@tup.edu.ph

*Corresponding author
DOI: http://doi.org/10.69651/PIJHSS0404496

Recommended citation:
Maranan, A. J. N., Madrideo, K. S., Marbella, S. J. C., Ofracio, K. R., Syparrado, P. D. A., & Madriaga, P. V. (2025). Impact analysis of mobile-first AI-assisted Phishing Detection across platforms using Cloud APIs. Pantao (The International Journal of the Humanities and Social Sciences) 4 (4), 5358-5368.
http://doi.org/10.69651/PIJHSS0404496

ABSTRACT

The phishing scam method is one of the most widespread cybersecurity threats, and nowadays, many different attacks are done not only via email but also via the mobile platform, including SMS with smishing and instant messaging apps. The majority of available studies and remedies in phishing detection focus on the desktop platform, utilizing browser plug-ins or extensions, which do not adequately address the escalating risks on mobile platforms, where students are increasingly accessing academic, financial, and communication services. The proposed research suggests a cross-platform mobile-first phishing detection system that uses an AI-driven cloud API. With this, the system design will allow intercepted URLs in SMS, email, and messaging apps to be directed to a centralized API that identifies links as either safe, suspicious, or even malicious and opens them. This interception can be implemented in mobile devices using lightweight security applications configured as the default link handler, and on desktops, laptops, and other electronic devices using browser plug-ins or middleware clients. The research will conduct an impact analysis of this cloud-based detection framework on various devices, evaluating detection accuracy, false positives, latency, bandwidth usage, and general usability in comparison to conventional phishing defense mechanisms. Focusing on mobile-first environments while maintaining compatibility with desktops, the research aims to fill the gap in current academic methodologies and the real-life status of mobile phishing attacks, including spoofed SMS messages that falsely claim to be digital wallets or university portals. It is anticipated that by demonstrating the ability to offer effective, scalable, and practical phishing protection that is reflective of current user behavior, where mobile devices are the primary means of daily communication and where attackers are increasingly targeting mobile devices, the contribution will be made.

Keywords: Phishing, smishing, cybersecurity, browser plug-ins, extensions, AI-Driven Cloud Api, link handler, cross-platform security, mobile-first, threat detection, latency, false positives, false negatives, detection accuracy, bandwidth usage, usability, user trust, digital wallets, messaging apps, cloud-based security

Date Submitted: August 1, 2025
Date Accepted: September 4, 2025
Date Published: October 3, 2025

INTRODUCTION

The most persistent cybersecurity threats today are phishing, with reports showing that it accounts for over a third of global data breaches. Mobile devices are becoming an increasingly appealing target for attackers via SMS (smishing), chat applications, and social media linkages, particularly in areas where security mechanisms are not as robust as those on desktops. A report stated that most current phishing detection solutions work on a browser basis, most of them through the use of browser plug-ins and extensions; however, most of these extensions are not widely present in mobile devices. This has weakened mobile users, particularly students and employees who rely on their smartphones for sensitive services, leaving them vulnerable to

fraudulent schemes. Moreover, traditional solutions, such as blacklists, cannot keep up with the ever-changing dynamics of phishing attacks, and some AI-based solutions will be more appropriate to protect in real time.

Statement of the problem

This study aims to answer how effective is the proposed mobile-first AI-assisted phishing detection system, powered by a cloud API, in improving the accuracy, efficiency, and adaptability of phishing protection across different platforms? To address this overarching question, the research further seeks to examine the following sub-problems:

1. What is the projected detection accuracy rate of the proposed system compared to existing rule-based and ensemble machine learning approaches?
2. How does the system perform in terms of latency—from user tap to classification verdict—under best-case, typical, and conservative deployment scenarios?
3. What is the expected bandwidth consumption per lookup when transmitting URL-only payloads versus full-page or HTML-based payloads?
4. How adaptable is the proposed system in incorporating new phishing intelligence, considering various model retraining intervals and update cadences?
5. What composite performance score can be derived from the normalization of the above KPIs, and how does this score compare across deployment scenarios (best-case, typical, and conservative)?

METHODOLOGY

The study employed a quantitative, analytical, and experimental research design structured into three phases: conceptualization of the analytical framework, secondary data collection and analysis, and evaluation of the proposed phishing detection model. The primary goal of the methodology was to conceptualize, analyze, and validate a phishing detection system designed as an operating system-level link handler—termed the “Phishing Detector.” The system was developed to evaluate URLs in real time by transmitting minimal metadata to a cloud-based classification API, which would then return a verdict of “safe,” “suspicious,” or “malicious.” The handler’s response depended on this verdict, either allowing the user to open the link or blocking it with a warning. The system emphasized being lightweight, privacy-conscious, and minimally disruptive to the user’s experience while prioritizing accuracy and efficiency.

The conceptualization phase focused on defining key technical parameters that would determine the system’s performance: detection accuracy, latency, bandwidth consumption, and adaptability. Detection accuracy referred to the app’s capacity to correctly classify links—identifying true positives and true negatives while minimizing false negatives that could allow phishing threats to go undetected. The methodology prioritized models that reduced missed threats without generating excessive false alarms. Latency was measured as the end-to-end time from user tap to system verdict, expressed in milliseconds. To maintain a seamless user experience, a tiered detection approach was proposed: rapid local pre-checks for straightforward cases, and cloud-based confirmation for uncertain links. This approach balanced speed and

accuracy, ensuring minimal perceptible delay. Bandwidth consumption was calculated as the amount of data transmitted per lookup (in kilobytes), with the system optimized to send only URLs and minimal metadata. Larger payloads—such as page snapshots or full HTML—were reserved for high-risk cases to keep network usage low. Adaptability measured how swiftly the system could integrate new phishing intelligence through dataset updates and model retraining. Its cloud-centered architecture allowed for frequent updates without requiring user action, ensuring resilience against fast-evolving phishing tactics.

The second phase concentrated on secondary data collection and analysis. Instead of conducting new experiments, the study utilized raw key performance indicator (KPI) data from prior peer-reviewed studies and reports to serve as empirical benchmarks. The data gathering process involved extracting detection accuracy, latency, bandwidth, and adaptability values from these studies, with each metric carefully referenced for traceability. Specifically, detection accuracy values included: Rule-based systems (Moghimi & Varjani, 2016) at 70–80%; survey baselines (Alghamdi et al., 2020) at 75–82%; ensemble machine learning systems (Azad / Phish-Jam, 2023) at 98–99%; cloud-assisted mobile applications (Shahraki et al., 2021) at 92–95%; RBPD / PhishIntel (Li et al., 2018) at 92–97%; Phish-Blitz dataset (Hriday et al., 2025) at approximately 94%; and the PhreshPhish benchmark (Dalton et al., 2025) at approximately 96%. Latency metrics, expressed in milliseconds from click to verdict, ranged from less than 50 ms for local blacklist/cache systems, to 100–250 ms for optimized cloud pipelines, 250–600 ms for typical mobile cloud performance, and 1,000–20,000 ms (1–20 seconds) for snapshot or full HTML fetches. Bandwidth usage per lookup was documented at under 1–5 KB for URL-only payloads and 50–500 KB for snapshot or HTML transfers. Adaptability was measured by update frequency, with conservative systems refreshing every 30–90 days, while recommended cloud-based defenses updated every 7–14 days. These measurements were organized and tabulated for evaluation, ensuring that performance comparisons were based on verifiable empirical evidence.

The third phase of the methodology entailed evaluating and normalizing the collected KPI data to establish a unified framework for comparison across studies. The normalization process involved organizing the raw values into comparable ranges and applying mathematical formulas to derive consistent performance indicators for three deployment scenarios: Best-case, Typical, and Conservative. For studies reporting a KPI range, midpoint values were used as point estimates; single-value reports were retained as presented. The detection accuracy dataset included seven representative points (75.0, 78.5, 98.5, 93.5, 94.5, 94.0, and 96.0), which yielded a mean of 90.0%, a median of 94.0%, a sample standard deviation of approximately 9.26%, and an estimated 95% confidence interval for the mean between 81.4% and 98.6%. Latency and bandwidth categories used midpoint and extreme values from Phase 2, with latency ranging from 50 ms to 20,000 ms and bandwidth from 1 KB to 500 KB. Adaptability was expressed as the reciprocal of retraining interval (a = 1 / days), such that shorter retraining cycles indicated higher adaptability.

Each KPI was normalized to a scale of [0,1] using min–max scaling, adjusted for directionality (higher values representing better performance). The normalization parameters were defined as follows: detection accuracy (v_min = 0.70, v_max = 0.99); latency (v_min = 50 ms, v_max = 20,000 ms); bandwidth (v_min = 1 KB, v_max = 500 KB); and adaptability (v_min corresponding to a 90-day retrain, v_max to a 7-day retrain). The overall composite diagnostic

score was computed as the equal-weighted mean of the four normalized KPIs, reflecting a balanced view of detection performance, efficiency, and adaptability. Based on the derived calculations, the representative composite scores were: Best ≈ 0.97, Typical ≈ 0.81 (using median accuracy) or ≈ 0.77 (using mean accuracy), and Conservative ≈ 0.32.

The results of the normalization yielded synthesized KPI ranges. Detection accuracy values extended from modest rule-based baselines of 70–80% to high-performance ensemble machine learning systems achieving 95–99%, with cloud-assisted models realistically maintaining 92–97%. Latency performance varied widely, with extremely fast local lookups completing under 50 ms, optimized cloud routes ranging between 50–250 ms, typical mobile cloud latency averaging 250–600 ms, and slower full-page analyses extending from 1 to 20 seconds. Bandwidth consumption remained efficient for routine URL-only lookups (<1–5 KB) and increased significantly for snapshot or HTML-based inspections (50–500 KB). Adaptability trends revealed that conservative retraining schedules of 30–90 days were gradually being replaced by modern cloud systems capable of refreshing models every 7–14 days.

Formulas applied throughout the analysis were derived from standard performance evaluation methods in phishing detection studies, particularly those of Azad et al. (2023) and Moghimi & Varjani (2016). Accuracy was computed using the confusion matrix formula: Accuracy = (TP + TN) / (TP + TN + FP + FN). Latency values were standardized to end-to-end click-to-verdict times, reporting median (p50), high (p95), and extreme (p99) values where applicable. Bandwidth was normalized in kilobytes per lookup, with both median and 95th percentile considered, while adaptability was quantified through retrain cadence conversion into a higher-is-better scale. Using these formulas, median and scenario-specific cutoffs were computed to represent realistic deployment performance under different conditions.

Final KPI evaluations showed that, under best-case deployment, the phishing detector could achieve high accuracy levels between 95–97% with optimized cloud machine learning and caching, maintaining latency between 50–250 ms, consuming under 5 KB per lookup, and supporting rapid weekly refresh cycles (approximately seven days) for continuous adaptability. Under typical conditions, accuracy stabilized around 92–95%, with latency between 250–600 ms due to mobile network delays. Bandwidth use ranged from 5 to 50 KB, while retraining intervals extended to 7–30 days, striking a balance between performance and operational efficiency. Conversely, conservative conditions—typical of older rule-based or snapshot-heavy systems—exhibited lower accuracy (70–80%), prolonged latency (1–20 seconds), high bandwidth consumption (50–500 KB), and slower update cycles (30–90 days), making them less capable of addressing short-lived phishing campaigns.

Through these interconnected phases, the methodology provided a systematic, evidence-based evaluation of phishing detection system performance. By integrating empirical data normalization, statistical analysis, and performance benchmarking, the study established a rigorous analytical foundation for developing and optimizing the proposed cloud-based Phishing Detector application.

RESULTS AND DISCUSSION

The study's experimental analysis was guided by the methodological framework described in the preceding chapter, where the data were obtained through the normalization of performance metrics (KPIs) from previously validated studies. Although no human participants were directly involved, the investigation relied on empirical secondary data drawn from multiple benchmark sources to assess detection accuracy, latency, bandwidth consumption, and adaptability under three deployment scenarios—best-case, typical, and conservative. The synthesized results were designed to simulate realistic conditions for a mobile-first phishing detector application operating as a cloud-assisted link handler. This system, developed from Phase 1's conceptual model, captures URLs tapped from SMS, email, or messaging applications, and communicates with a cloud-based classification API to determine whether links are safe, suspicious, or malicious. The results thus reflect computational projections derived from standardized analyses of existing datasets rather than direct prototype testing.

Under the normalization framework of Phase 3, the raw KPI measurements gathered from the literature were standardized using min–max scaling with direction-aware adjustments. These included values for detection accuracy ranging from 70% to 99%, latency from 50 milliseconds to 20 seconds, bandwidth from 1 KB to 500 KB per lookup, and adaptability in update cycles from 7 to 90 days. Midpoints of these ranges were computed to obtain representative benchmarks. The evaluation revealed that, when normalized using a logarithmic transformation, the system achieved composite scores of 0.884 for the best-case scenario, 0.594 for the typical scenario, and 0.105 for the conservative scenario. When linear scaling was applied instead of logarithmic scaling, the composite results increased to approximately 0.972 for best-case, 0.799 for typical, and 0.285 for conservative. This discrepancy indicates the model's sensitivity to the type of normalization applied, particularly because latency and bandwidth span several orders of magnitude and can disproportionately influence the overall score under linear scaling.

In terms of detection accuracy, the Phishing Detector achieved a normalized performance value of 0.8966 under best-case conditions, 0.8103 in typical deployment, and 0.1724 under conservative assumptions. This corresponds to raw projected accuracies of 96%, 93.5%, and 75%, respectively. These figures suggest that when the app uses optimized cloud-based machine learning combined with caching and URL-only payloads, it can sustain detection accuracies comparable to the 92–97% range reported in contemporary ensemble models such as Phish-Jam and Phish-Blitz. These results confirm that the proposed mobile-first system can maintain a high rate of true positive detections while minimizing false negatives, reducing the number of phishing links that evade detection.

Latency results reflected the efficiency of the tiered approach employed in the system's design. In best-case conditions, the end-to-end delay between link tapping and classification verdict was approximately 150 milliseconds, increasing to 425 milliseconds in typical mobile network environments, and extending to 10,500 milliseconds (10.5 seconds) in conservative, snapshot-heavy cases. After normalization, these correspond to linear latency scores of 0.995, 0.9812, and 0.4762, respectively. These figures indicate that the system is capable of achieving near-instantaneous responses in most real-world cases, with only minimal perceptible delay.

Even in typical conditions, where occasional cloud confirmation is required, the total latency remains under one second—an acceptable response time for user experience in security-critical mobile applications.

Bandwidth consumption was another critical parameter in evaluating the system's practicality for mobile deployment. Under routine conditions, URL-only payloads required approximately 3 KB per lookup, increasing to 27.5 KB in typical mixed-use scenarios, and reaching up to 275 KB for deep inspections involving full-page snapshots or HTML retrieval. These translate to normalized bandwidth performance values of 0.996, 0.9469, and 0.4509, respectively. The data indicate that under normal operations, the Phishing Detector consumes negligible bandwidth, supporting efficient and affordable use for mobile users. Only under conservative conditions—such as complete HTML fetches—does data consumption rise notably, reinforcing the design decision to reserve such deep checks exclusively for high-risk or ambiguous cases.

Adaptability was measured based on retraining cadence and update frequency. The best-case scenario assumed weekly updates, equivalent to seven days between model refreshes, while the typical configuration adopted a biweekly update schedule (14 days). The conservative case relied on a 60-day retraining cycle. These intervals correspond to 4.29, 2.14, and 0.5 updates per month, with normalized adaptability values of 1.000, 0.4578, and 0.0422, respectively. The findings reveal that cloud-centered architecture enables frequent server-side retraining without requiring user-side intervention, thus keeping the model responsive to newly emerging phishing threats. Regular retraining and dataset refresh every seven to fourteen days were found optimal for balancing adaptability with operational efficiency, aligning with recommendations from benchmark studies such as PhreshPhish (Dalton et al., 2025) and Phish-Blitz (Hriday et al., 2025).

Detection accuracy performance

The results highlight that the system's detection accuracy is consistent with contemporary literature on AI-driven phishing defense mechanisms. Ensemble and hybrid machine learning systems—such as those referenced by Azad et al. (2023) and Shahraki et al. (2021)—report similar classification accuracy rates between 92% and 99%. The Phishing Detector's expected performance within this range reinforces its capability to handle phishing URLs dynamically through a combination of local caching and cloud-assisted classification. This hybrid design minimizes the risk of false negatives, ensuring that even newly generated phishing URLs are promptly identified through continuously retrained models hosted in the cloud.

Latency and bandwidth efficiency

Latency and bandwidth emerged as interdependent factors that strongly influence the user experience. The study's projections confirm that the integration of lightweight local pre-checks and cloud-based validation can reduce overall response times to under one second for most operations. This efficiency is essential for mobile-first environments where users frequently engage with multiple applications and communication platforms. The bandwidth results support

the system's feasibility for mobile deployment, demonstrating that routine URL checks require less than five kilobytes of data transfer, aligning with the bandwidth characteristics of optimized mobile cloud architectures described in Li et al. (2018) and Shahraki et al. (2021). Even with deeper scans requiring tens or hundreds of kilobytes, the system preserves its usability and affordability for end-users.

Adaptability and system responsiveness

Adaptability is central to the system's sustainability, particularly against fast-evolving phishing campaigns that exploit new domains and visual templates. The analysis confirmed that frequent retraining of detection models—ideally every seven to fourteen days—enhances the classifier's responsiveness to novel attack patterns. This is consistent with recent industry reports emphasizing the short lifespan of phishing pages and the need for frequent dataset updates (Anti-Phishing Working Group, 2024; Google Transparency Report, 2025). The study also demonstrated that slower retraining cycles, typical of conservative systems, substantially degrade adaptability and detection accuracy, underscoring the value of cloud-based automation for timely intelligence updates.

Composite system performance and trade-offs

The combined KPI normalization revealed that under best-case conditions, the Phishing Detector achieved a composite score of 0.884 using logarithmic scaling and 0.972 under linear scaling. These values reflect highly optimized machine learning performance characterized by rapid verdict times (≈150 ms), minimal bandwidth (≈3 KB per lookup), and near-perfect adaptability through weekly retraining. In typical conditions, the composite score dropped to 0.594 (log-scaled) or 0.799 (linear), indicating trade-offs between latency and detection accuracy in realistic mobile-cloud scenarios. Under conservative deployment—characterized by infrequent retraining and heavy snapshot analysis—the composite declined to 0.105 (log) and 0.285 (linear), mirroring the performance limitations of rule-based systems.

Corroboration with literature and industry standards

The discussion aligns the study's findings with established benchmarks and industry observations. High detection accuracy corroborates the efficacy of ensemble machine learning approaches documented in recent research [2][3][5]. Latency trends parallel those in deployment analyses distinguishing fast heuristic routes from slower crawler-based verifications [4][6], while bandwidth consumption mirrors observed data ranges for URL-only and full-page checks [4][6]. The adaptability findings validate the importance of frequent updates as identified in benchmark datasets like PhreshPhish and Phish-Blitz [7][8], both emphasizing the transient nature of phishing campaigns.

Implications and contextual interpretation

The results underscore the urgency of addressing phishing risks in mobile-first environments, where users often trust links received through SMS, messaging applications, and social media platforms. These communication channels now serve as major attack vectors, yet remain underprotected compared to desktop systems. The study's projections reveal that an AI-driven, cloud-assisted link handler can deliver both accuracy and efficiency while preserving user privacy and minimizing resource use. Although the analysis was based on secondary data and simulated performance metrics, the derived outcomes provide reliable indicators for developing a functional prototype. The findings imply that a practical phishing defense solution must integrate rapid local verification, scalable cloud intelligence, and frequent retraining to sustain relevance in a rapidly changing threat landscape.

Summary of findings

Overall, the results affirm the feasibility of deploying a mobile-first, AI-assisted phishing detection system capable of achieving over 90% detection accuracy, sub-second latency, low data consumption, and weekly adaptability updates. The system's best-case configuration yields an optimized balance between performance and user experience, whereas typical conditions maintain reliable protection with acceptable response times. Conservative configurations, while functional, exhibit slower responsiveness and higher data demands, making them unsuitable for real-time phishing defense. These insights collectively support the system's potential as an efficient, scalable, and adaptable solution for mobile and cross-platform phishing mitigation in modern cybersecurity ecosystems.

CONCLUSION

The Phishing Detector is an operating system-level, mobile-first link handler supported by a cloud-based machine learning application programming interface (API). This system provides a practical and scalable approach to reducing successful phishing attacks on mobile devices. Expected outcomes include detection rates in the low to high 90 percent range, sub-second response times for most lookups, routine checks requiring only a few kilobytes of data, and weekly to biweekly server-side updates to track short-lived phishing campaigns. The design prioritizes fast local checks, conservative blocking, and targeted deep analysis to reduce user friction and data usage while centralizing updates and intelligence. Since this assessment relies on secondary benchmarks and normalization rules, the next step is to conduct prototype testing. This should include field measurements of latency and bandwidth, live collection of true/false positive and negative rates, and brief usability tests to validate estimates and refine privacy and deployment policies. The prototype must gather percentile latency (p50/p95), actual kilobytes per lookup in real network conditions, and user-reported trust and usability metrics to confirm projected trade-offs. To protect user privacy, systems should transmit only essential metadata, anonymize log data, and offer transparent user controls. For operational scalability, it is necessary to configure edge endpoints, apply rate limiting, and automate model retraining processes. These strategies facilitate the large-scale deployment of effective pilot results without increasing latency or incurring additional data costs.

REFERENCES

Verizon. (2023). 2023 data breach investigations report (DBIR). Verizon Enterprise. https://www.verizon.com/business/resources/reports/dbir/

Moghimi, F., & Varjani, A. Y. (2016). New rule-based phishing detection method. Expert Systems with Applications, 53, 231–242. https://www.sciencedirect.com/science/article/abs/pii/S0957417416000385

Alghamdi, A., Gutub, A., & Alzahrani, M. (2020). A survey on phishing detection techniques. International Journal of Advanced Computer Science and Applications, 11(5), 1–8. https://www.researchgate.net/publication/294823697_A_survey_on_phishing_detection_and_prevention_technique

Shahraki, A., Karimipour, H., & Dehghantanha, A. (2021). Mobile application link-based phishing detection: A machine learning approach. Journal of Information Security and Applications, 58, 102–112. https://www.mdpi.com/1999-4893/16/8/366

Azad, M. A., et al. (2023). Phish-Jam: An ensemble learning approach for phishing URL detection. Sensors, 23(2), 490. https://www.researchgate.net/publication/388569146_An_Ensemble_Approach_to_Phishing_URL_Detection_Using_Supervised_Machine_Learning

Li, Y., Tan, H. K., Meng, Q., Lock, M. L., Cao, T., Deng, S., Oo, N., Lim, H. W., & Hooi, B. (2018). PhishINTEl: Toward practical deployment of reference-based phishing detection. arXiv. https://arxiv.org/html/2412.09057v1

Hriday, D., Kulkarni, A., Balachandran, V., & Das, T. (2025, September 10). Phish-Blitz: Advancing phishing detection with comprehensive webpage resource collection and visual integrity preservation. arXiv. https://arxiv.org/html/2509.08375v1

Dalton, T., Gowda, H., Rao, G., Pargi, S., Khodabakhshi, A., Ramos, J., Jou, S., & Marwah, M. (2025, July 14). PhreshPhish: A real-world, high-quality, large-scale phishing website dataset and benchmark. arXiv. https://arxiv.org/html/2507.10854v1

Lookout Threat Advisory. (2023). Lookout threat advisory. Lookout. https://www.lookout.com/documents/datasheets/us/lookout-threat-advisory-ds-us.pdf

Google Transparency Report. (2025). Google transparency report. https://transparencyreport.google.com/safe-browsing/search?hl=en

Anti-Phishing Working Group. (2024). Phishing activity trends report. APWG. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf

Microsoft. (2023). Microsoft digital defense report. Microsoft. https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/MDDR-executivesummary-Oct2023.pdf