Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2025.0322000

Digital Wolves in Sheep's Clothing: Detecting Malicious Domains

RADEK HRANICKÝ[1], JAN POLIŠENSKÝ[1], ONDŘEJ ONDRYÁŠ[1], KAMIL JEŘÁBEK[1],
[1]Faculty of Information Technology, Brno University of Technology, Božetěchova 2/1, 612 00 Brno, Czechia (e-mail: author@boulder.nist.gov)

Corresponding author: Radek Hranický (e-mail: hranicky@ fit.vut.cz).

The research presented in this paper was supported by ''Flow-based Encrypted Traffic Analysis'' project, no. VJ02010024 granted by the Ministry of the Interior of the Czech Republic and ''Smart information technology for a resilient society'' project, no. FIT-S-23-8209 granted by Brno University of Technology.

ABSTRACT Detecting malicious domains in real-time remains a critical challenge in cybersecurity, particularly in high-volume DNS traffic environments where millions of domain names must be processed daily. This paper presents a modular and adaptive multi-stage classification pipeline for detecting phishing and malware-related domains, designed with a focus on scalability, efficiency, and practical deployability. Unlike prior methods that rely on complete feature sets for all inputs, our approach begins with lightweight lexical analysis and selectively deepens the analysis only for domains that exhibit signs of suspiciousness. This enables substantial resource savings while maintaining high detection accuracy. The system is evaluated on a large real-world dataset collected from a national-level ISP, as well as on an isolated test set gathered during a different time period to assess generalization to unseen data. We also replicate several state-of-the-art malicious domain detection methods and compare them against our approach. Our pipeline outperforms all of them, achieving an F1 score of 0.985 for phishing domain detection and 0.980 for malware-related domain detection. The results demonstrate that the proposed pipeline combines strong classification performance with efficient data collection and processing, thereby offering a viable solution for large-scale deployment in operational cybersecurity environments.

INDEX TERMS Phishing, Detection, Classifier, Features, Lexical, DNS, RDAP, TLS, GeoIP

I. INTRODUCTION

The domain name system (DNS) plays a central role in modern internet communication, making it a frequent target and facilitator of cyber attacks. Malicious domains are routinely employed in a wide spectrum of threats, such as phishing campaigns designed to steal user credentials, command-and-control (C&C) infrastructures for managing botnets, and the delivery or distribution of malware. Detecting these domains early, ideally before any damage occurs, is a crucial component of any robust cybersecurity strategy. Traditional defenses, such as static blacklists and handcrafted heuristics, are increasingly insufficient in the face of sophisticated adversaries who can generate large volumes of novel domains using automation techniques like domain generation algorithms (DGAs), evade detection through rapid domain rotation, or exploit unclassified infrastructure. As a result, recent research has focused on machine learning-based approaches that identify malicious domains by analyzing structural and behavioral features extracted from domain names and associated metadata.

Despite their promise, existing machine learning methods face significant barriers to deployment in realistic, high-throughput environments. Many rely on comprehensive feature vectors derived from external data sources such as WHOIS, RDAP, DNS records, or TLS certificates, which may be unavailable, unreliable, or subject to strict rate limiting. These models typically assume full data availability and apply the same processing path to all domains, regardless of their initial risk profile. In practice, cybersecurity systems must process millions of domain names each day. Collecting and evaluating full-feature information for every domain would lead to unacceptable delays and high resource consumption. Furthermore, some prior studies evaluate their models only on static or synthetic datasets, without validating performance on real-world traffic or accounting for temporal drift, which limits their practical relevance.

In this paper, we propose and evaluate a practical, multi-stage domain classification pipeline that addresses the challenges of large-scale, real-time detection. The system begins with lightweight lexical analysis of domain names, which requires no external data and can quickly rule out clearly benign cases. Only domains that appear suspicious in this initial phase proceed to further stages of enrichment, where additional metadata such as DNS resolution details, IP ownership, RDAP attributes, or TLS fingerprinting is collected and analyzed. This adaptive depth strategy allows the system to balance detection accuracy with resource efficiency, focusing computational effort on high-risk cases while minimizing overhead on benign traffic. We implement the full pipeline and evaluate it using real-world data captured from a national ISP environment. Our results demonstrate that the proposed approach supports scalable deployment and outperforms several state-of-the-art baselines in terms of classification performance, achieving an F1 score of 0.985 for phishing detection and 0.980 for malware-related domain detection.

A. RESEARCH GOALS

In this work, we aim to bridge the gap between academic research in malicious domain detection and their practical deployment in real-world environments. Unlike many prior methods that assume full feature availability and unrestricted data collection, our focus is on designing a solution that remains effective under operational constraints such as rate limits, incomplete data, and the need for real-time processing. Concretely, our goals were to:

G.1 Design a modular and adaptive domain classification pipeline for detecting phishing and malware-related domains with high accuracy under real-world constraints. The pipeline should operate efficiently on high volumes on domain names, avoid unnecessary data collection, and utilizing lightweight models in early decision stages.

G.2 **[TODO: upravit stylem "vytvořit vhodný dataset na základě reálných dat")]** Validate the system using real traffic collected from a national-level ISP environment to ensure its applicability under realistic operational conditions.

G.3 Evaluate the system's generalization capabilities by testing it on a completely isolated real-world dataset of domain names, obtained during a different time period, not used during training. **[TODO: přidat "national-level ISP environment" sem]**

G.4 Compare the system against previously published methods by reproducing them on our datasets and demonstrating comparative improvements in classification performance.

B. CONTRIBUTION

This article presents the following key contributions:

- We introduce a novel multi-stage domain classification architecture that dynamically adjusts feature usage based on the domain name's initial risk score and available metadata. The system begins with fast, lexical-only analysis and proceeds to more data-intensive stages only if necessary.
- Our system is designed for real-world deployment, prioritizing efficiency and practical applicability over theoretical complexity. It enables large-scale analysis of millions of domain names daily without requiring full data collection for every input.
- We perform extensive experimental evaluation on a large dataset of real-world DNS traffic, including a fully isolated test set collected during a different time window to simulate deployment on unseen data.
- We reproduce and benchmark several prior state-of-the-art approaches using our data and demonstrate that our method outperforms all known models with publicly available feature sets. We achieve an F1 score of 0.985 for phishing detection and 0.980 for malware-related domain detection, setting a new baseline in this field.

C. STRUCTURE OF THE PAPER

The rest of the paper is organized as follows. Section II reviews related work in domain-based threat detection and highlights the limitations of existing approaches. Section III describes the dataset used for evaluation, including the real-world DNS capture and its segmentation. Section IV outlines the methodology and the design of the multi-stage classification pipeline. Section V presents the experimental setup and results, including performance metrics and a comparison with prior work. Section VI discusses the implications of our findings and deployment considerations. Finally, Section VII concludes the paper and outlines directions for future research.

II. RELATED WORK

[TODO: Kamil: rozpracovat a dokončit, přidat malware-domain related články]

Throughout the history, security operators have looked for different ways to fight against malicious threats on the internet including phishing, C2C or malware spread communication. They can be divided into several categories as specified by Khonji et al. [1] utilizing different aspects of network communication including domain names. User awareness is the first technique relying on the user training. Several studies [2]–[4] proved that this technique fails in many caases even with expert individuals influence by human emotions. Known user failure rates are prevented by other sophisticated widely used methods including simple allow/deny lists that are commonly implemented by various systems such as DNSBL [5] in DNS servers, providing effective method against malicious threats as shown in works [6]–[8]. However, they are known to be prone to miss zero-hour attacks [8] as they usually rely on delayed user reported domains.

Therefore, more sophisticated methods were proposed. At first, heuristics crafted to capture communication or behavioral patterns were developed [1], [8]–[10]. Later, machine learning was proposed by many works [11]–[13] to extract and learn common patterns from high amounts of different data providing a viable method for identifying current and future threats.

III. DATASET AND DATA ANALYSIS

[TODO: OO revise] The system is built on a dataset of over one million labeled domains spanning benign, phishing, and malware classes. Metadata enrichment was a critical part of the architecture, enabling precise segmentation and informed classification. The data collection process was designed for scalability, precision, and fault tolerance, and was executed using custom modular Python pipelines.

A. DATASET COLLECTION PROCESS

[TODO: kecy kecy – vycuc z DiB] [TODO: ref na OO DP?]

a: Source Aggregation and Labeling.

Benign domains were collected from long-term snapshots of the Cisco Umbrella Top 1M list and TLS SNI traffic observed within the CESNET academic backbone. To reduce potential mislabeling, only domains with persistent presence over a 12-month period were retained. Malicious domains were ingested from continuously updated threat intelligence feeds, including PhishTank, OpenPhish, URLHaus, and Abuse.ch. A local MISP instance handled ingestion, deduplication, and time-window filtering.

b: VirusTotal Validation.

To ensure high label fidelity, all domains—especially those labeled as malicious—were revalidated through the VirusTotal academic API. Only domains confirmed by multiple security vendors were retained in the malicious set. Conversely, benign domains with any detection flags were excluded. This validation step significantly improved dataset quality and reduced the likelihood of false positives.

c: Metadata Enrichment.

Following initial collection, each domain was passed through a custom Enrichment Data Collector. The collector retrieved:

- DNS records: A, AAAA, NS, MX, TXT, SOA, DNSSEC
- IP-based data: GeoIP location, ASN, and CESNET RTT measurements
- TLS handshake and X.509 certificate details
- RDAP/WHOIS records

Enrichment was performed in incremental best-effort rounds, with retries for failed queries. All metadata was stored in a unified MongoDB schema and exported to structured JSON for downstream processing. This modular design ensured resilience to outages and third-party throttling.

B. DATASET ANALYSIS

[TODO: OO] Praesent est nisi, posuere ac urna vitae, ullamcorper eleifend odio. Integer venenatis ligula nibh, a consectetur velit elementum sit amet. Nulla semper convallis purus nec vulputate. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam porta leo nec odio venenatis laoreet. Donec commodo nisi vel erat faucibus, ut dignissim erat efficitur. Proin laoreet sem ex, vulputate sollicitudin lorem laoreet at. Nulla finibus nec nibh sit amet molestie. Sed elementum id dolor et cursus.

IV. METHODOLOGY

A. MULTI-STAGE CLASSIFICATION DESIGN

In real-world phishing detection systems, the availability and quality of metadata can vary dramatically across domain samples due to factors such as DNS resolution errors, inaccessible registrars, rate-limiting of RDAP servers, or incomplete TLS handshakes. [TODO: viz kap. III] Relying on a single uniform feature set would result in either discarding a large portion of domains due to missing data or degrading model performance by ignoring valuable contextual features when available.

To overcome this, our system employs a multi-stage classification architecture. Domains are routed into distinct processing branches based on the actual availability of auxiliary data (e.g., DNS, IP, TLS, RDAP), allowing each classifier to operate on the richest possible feature set without introducing null-padding or imputation noise. This design avoids degrading the model's decision boundaries due to inconsistently populated input vectors while maximizing data utilization.

By separating samples into stages, we reduce overfitting and improve interpretability — each classifier is trained and validated on a consistent subset of features, better reflecting the realistic operating conditions it will encounter. Moreover, segmentation ensures fail-safe execution: no domain is discarded solely due to missing metadata. The system always falls back to a more lightweight stage, preserving coverage.

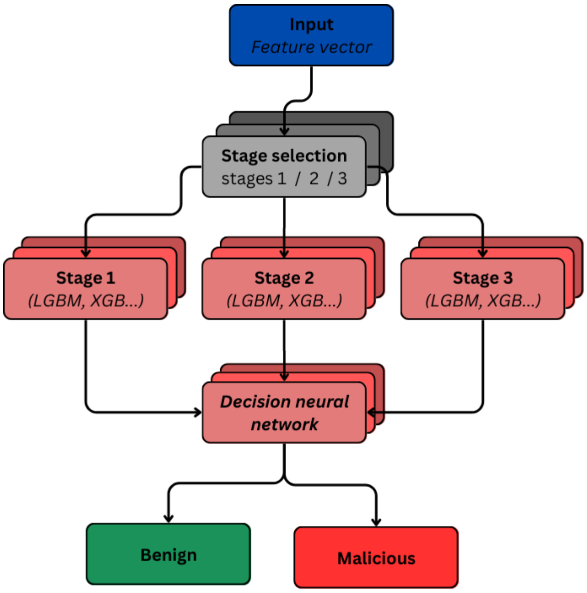

FIGURE 1. Overview of the multi-stage classification pipeline with decision network.

a: Within-Stage Decision Strategies and Weighting Mechanisms

[TODO: Poli: Toto celé přesunout až "na konec" – první bude výběr stages, pak povídání o modelech uvnitř stage, nakonec agregace těch výsledků z modelů. Předělat úrovně nadpisů, aby to mělo logickou strukturu.]

The outputs from models within each stage (e.g., LightGBM, XGBoost) are aggregated using several possible strategies:

- **Best model selection:** Only the output of the most accurate model (based on validation F1-score) within the stage is passed forward.
- **Unweighted average:** The final output is computed as the arithmetic mean of all individual model outputs in the stage.
- **Weighted average:** A weighted mean is used, where each model's output is scaled by its validation F1-score, giving more influence to stronger models.
- **Majority voting:** The final label is decided by majority vote among all model predictions (suitable for binary classification).
- **Bayesian aggregation:** Posterior probabilities are estimated from model outputs using a naïve Bayes assumption across classifiers.
- **Decision neural network (meta-classifier):** A feed-forward neural network takes the stage outputs (probabilities or logits) as input and learns to combine them optimally.

The final architecture thus integrates both hard-coded strategies (e.g., averaging or voting) and learned decision logic (neural meta-classifier). The latter was particularly beneficial in handling edge cases and reducing false positives in ambiguous samples.

b: Comparison of Aggregation Strategies

To assess the impact of different aggregation strategies, we evaluated their classification performance at Stage 3. The results are summarized in Table 1.

Method	Accuracy	Precision	Recall	F1-score
Best model	0.9952	0.9927	0.9790	0.9858
Unweighted average	0.9947	0.9926	0.9757	0.9841
Weighted average	0.9947	0.9926	0.9757	0.9841
Meta-model (NN)	0.9953	0.9928	0.9792	0.9860

TABLE 1. Performance comparison of ensemble aggregation strategies (Stage 3).

Although all ensemble methods achieved strong performance, the meta-model and best-model strategies delivered the highest F1-score of 0.9860, indicating effective balance between precision and recall. However, the meta-model provides an additional advantage: it can adapt its weighting scheme dynamically to each input sample based on learned decision patterns, improving robustness in noisy and ambiguous cases. Given its comparable performance and superior flexibility, the neural network meta-classifier was selected as the final decision layer in the system.

B. SEGMENTATION PIPELINE

[TODO: OO: navazat na kap. III]

a: Routing and Segmentation.

The collected metadata was used to dynamically assign each domain to one of three classification stages:

- **Stage I:** Lexical-only (domain name string features)
- **Stage II:** DNS + IP + geolocation (middle-tier metadata)
- **Stage III:** Full enrichment including TLS and RDAP

This segmentation ensures that no sample is discarded due to missing features. Instead, every domain is classified with the best available context. This adaptive routing mimics realistic operational conditions where feature completeness cannot be guaranteed.

[TODO: Poli: Jak jsi přišel na to, že to má smysl dělat zrovna takhle?]

1) Stage I: Lexical-Only Classification

The first classification stage uses only lexical features derived directly from the domain name string. These include statistical, structural, and linguistic properties of the domain, and they are always available regardless of external data sources. This stage serves as a fallback for cases where DNS resolution or metadata retrieval fails, allowing minimal yet nontrivial classification **[TODO: odkazy na nějakou related work, kde ukazují, že to jde?]**.

2) Stage II: DNS, IP and Geolocation-Enriched Features

If a domain resolves to a valid IP address, the second classification stage is activated. It extends the feature set with DNS response data, IP-level attributes, and derived geolocation information. These features can typically be collected quickly and reliably **[TODO: jak to víme?]**, and they significantly enhance the context of a domain without requiring deep inspection. This stage captures middle-tier domains where registration or encryption data is unavailable.

3) Stage III: Full Metadata with TLS and RDAP

The third and final stage leverages full metadata availability, including TLS certificate details and RDAP (Registration Data Access Protocol) records. These features provide insights into the identity of domain owners, certificate issuers, and other structural or administrative patterns associated with the domain. While more expensive to retrieve, they offer the richest analytical power and enable the most precise classification.

Routing Logic Based on Data Availability

Each domain is dynamically assigned to the most complete classification stage supported by its available data. If TLS or RDAP queries fail, but DNS and IP geolocation succeed, the domain is processed by Stage II. If even DNS resolution is unsuccessful, the domain is evaluated in Stage I using only lexical features. This adaptive mechanism ensures robust handling of real-world conditions without discarding incomplete records.

Classifier Selection and Integration

[TODO: Poli. Rozvinout.] For each stage, classifiers were selected using PyCaret to benchmark multiple algorithms. The final model ensemble includes gradient boosting (XGBoost, LightGBM), support vector machines (SVM), and neural networks (FFNN, CNN). These base models operate in parallel and their outputs are aggregated by a neural meta-classifier trained to synthesize their predictions into a final decision. This architecture allows both performance and resilience across diverse feature subsets.

Model	Stage	Accuracy	F1	ROC AUC
XGBoost	Stage 1	0.9651	0.8884	0.9829
LightGBM	Stage 1	0.9529	0.8446	0.9706
FFNN	Stage 1	**0.9676**	**0.8942**	**0.9838**
SVM	Stage 1	0.9630	0.8840	0.9389
XGBoost	Stage 2	0.9782	0.9328	0.9951
LightGBM	Stage 2	**0.9880**	**0.9638**	**0.9982**
FFNN	Stage 2	0.9801	0.9404	0.9643
SVM	Stage 2	0.9801	0.9801	0.9801
XGBoost	Stage 3	**0.9953**	**0.9860**	**0.9996**
LightGBM	Stage 3	0.9905	0.9711	0.9987
FFNN	Stage 3	0.9927	0.9780	0.9857
CNN	Stage 3	0.9546	0.9654	0.9706
SVM	Stage 3	0.9715	0.9691	0.9891

TABLE 2. Metrics for different models and stages for classification pipeline

C. EVALUATION

[TODO: Poli: Jakým způsobem bude provedeno vyhodnocení v kap. V?]

Curabitur feugiat felis et eros efficitur ornare. Quisque non dignissim velit. Quisque sit amet nulla tortor. Nulla iaculis, augue a feugiat egestas, purus eros aliquet dolor, id gravida felis neque a ligula. Morbi vitae hendrerit enim. Donec vitae fringilla risus, ac feugiat magna. Nulla tellus nisi, varius sed bibendum nec, laoreet at mauris. Praesent imperdiet, ex nec maximus pharetra, massa felis finibus massa, eget rutrum lectus tortor vel purus. Etiam quis sem tellus. Duis facilisis lectus quis mattis suscipit.

V. EXPERIMENTAL RESULTS

1) **[TODO: Srovnání s ostatními]**
2) **[TODO: Metriky na testovací sadě]**
3) **[TODO: Metriky na nezávislém datasetu z jiného období (clftest)]**
4) **[TODO: Velikost modelů, čas trénování, rychlost klasifikace na stejném HW (pro ty finálně vybrané modely)]**

VI. DISCUSSION

[TODO: Všichni, dopsat diskuzi.]

1) **[TODO: Degradace v čase]**
2) **[TODO: Rozšiřování FV? Změna struktury i obsahu dat?]**

First Author	Year	cat.	Model	F1 Phish.	F1 Malw.
Shi	2017	MIX	LightGBM	0.915	0.915
Torroledo	2018	TLS	LightGBM	0.905	0.966
Zhu	2019	MIX	AdaBoost	0.910	0.910
Magalhães	2020	MIX	LightGBM	0.969	0.969
Chatterjee	2019	MIX	XGBoost	0.924	0.900
Gopinath	2020	MIX	LightGBM	0.915	0.937
Hason	2020	MIX	LightGBM	0.971	0.971
Sadique	2020	MIX	XGBoost	0.924	0.900
Silveira	2021	DNS	SVM	0.921	0.921
Iwahana	2021	MIX	LightGBM	0.968	0.968
Kumar	2022	LEX	AdaBoost	0.890	0.932
This article	2025	MIX	Ensemble	**0.985**	**0.980**

TABLE 3. Comparison with related studies (Phish. – phishing dataset, Malw. – malware dataset) on the test portion of each dataset

First Author	Year	cat.	Model	F1 Phish.	F1 Malw.
Shi	2017	MIX	LightGBM	0.891	0.888
Torroledo	2018	TLS	LightGBM	0.881	0.945
Zhu	2019	MIX	AdaBoost	0.890	0.887
Magalhães	2020	MIX	LightGBM	0.950	0.946
Chatterjee	2019	MIX	XGBoost	0.900	0.877
Gopinath	2020	MIX	LightGBM	0.894	0.915
Hason	2020	MIX	LightGBM	0.954	0.950
Sadique	2020	MIX	XGBoost	0.899	0.880
Silveira	2021	DNS	SVM	0.894	0.891
Iwahana	2021	MIX	LightGBM	0.948	0.945
Kumar	2022	LEX	AdaBoost	0.867	0.910
This article	2025	MIX	Ensemble	**0.963**	**0.956**

TABLE 4. Comparison on an isolated dataset (5,048 domain names). Lower F1-scores reflect greater variance and reduced generalization.

3) ... ???

Curabitur feugiat felis et eros efficitur ornare. Quisque non dignissim velit. Quisque sit amet nulla tortor. Nulla iaculis, augue a feugiat egestas, purus eros aliquet dolor, id gravida felis neque a ligula. Morbi vitae hendrerit enim. Donec vitae fringilla risus, ac feugiat magna. Nulla tellus nisi, varius sed bibendum nec, laoreet at mauris. Praesent imperdiet, ex nec maximus pharetra, massa felis finibus massa, eget rutrum lectus tortor vel purus. Etiam quis sem tellus. Duis facilisis lectus quis mattis suscipit.

VII. CONCLUSION

Suspendisse odio nunc, condimentum at sollicitudin vel, blandit quis purus. Nunc dictum volutpat ex non malesuada. Fusce finibus fringilla lorem vel finibus. Suspendisse ut ligula lobortis, placerat nulla sed, faucibus libero. Phasellus non purus a ex pellentesque aliquet eu sit amet lectus. Aliquam vitae lacus mi. Praesent bibendum tempor diam, nec porttitor est porta sit amet. Quisque ac lorem sem. Quisque sollicitudin venenatis risus eu ultricies. Mauris sit amet efficitur risus. Etiam gravida sodales felis vel tincidunt.

APPENDIX A FOOTNOTES

Number footnotes separately in superscript numbers.[1] Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table **??**).

ACKNOWLEDGMENT

The research presented in this paper was supported by "Flow-based Encrypted Traffic Analysis" project, no. VJ02010024 granted by the Ministry of the Interior of the Czech Republic and "Smart information technology for a resilient society" project, no. FIT-S-23-8209 granted by Brno University of Technology.

REFERENCES

[1] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[2] 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.

[3] S. Gorling, "The Myth of User Education," in *Proceedings of the 16th Virus Bulletin International Conference*, Montreal, Canada, October 2006.

[4] B. A. Gyunka and A. O. Christiana, "Analysis of human factors in cyber security: A case study of anonymous attack on HBGary." *Computing & Information Systems*, vol. 21, no. 2, 2017.

[5] C. Lewis and M. Sergeant, "Overview of Best Email DNS-Based List (DNSBL) Operational Practices," RFC 6471 (Informational), Internet Engineering Task Force (IETF), Request for Comments (RFC) 6471, Jan. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6471.txt

[6] S. Bell and P. Komisarczuk, "An analysis of phishing blacklists: Google safe browsing, openphish, and phishtank," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2020, pp. 1–11.

[7] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–5.

[8] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 2009.

[9] D. L. Cook, V. K. Gurbani, and M. Daniluk, "Phishwish: a stateless phishing filter using minimal rules," in *Financial Cryptography and Data Security: 12th International Conference, FC 2008, Cozumel, Mexico, January 28-31, 2008. Revised Selected Papers 12*. Springer, 2008, pp. 182–186.

[10] E. Kirda and C. Kruegel, "Protecting users against phishing attacks with antiphish," in *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*, vol. 1, 2005, pp. 517–524 Vol. 2.

[11] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *NYS cyber security conference*, vol. 3. Albany, New York, 2006, pp. 2–8.

[12] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 649–656.

[13] 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.

[1] It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date on the first page). Instead, try to integrate the footnote information into the text.