OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 03 - Issue 05 (2026)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

AGENT-MD: A HUMAN-GOVERNED AGENTIC AI FRAMEWORK FOR EXPLAINABLE AND UNCERTAINTY-AWARE MALICIOUS DOMAIN DETECTION

Bushra Shaikh[1], Khakoo Mal[2], Noor Ahmed Shaikh[3], *Nizamuddin Maitlo[4]

[1, 3, 4]Faculty of Physical Sciences, Institute of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan.

[2]Department of Computer Science, Sukkur IBA University, Sukkur, Sindh, Pakistan.

[*]Corresponding Author: (nizamuddin.cs@gmail.com)
DOI:(*https://doi.org/10.71146/kjmr935*)

Article Info

Abstract

Malicious domains remain a durable part of the infrastructure used for phishing, malware distribution, command-and-control communication, spam campaigns, and online fraud. Prior work on malicious-domain and malicious-URL detection has shown that lexical, structural, DNS, host, and content-derived features can support accurate machine learning models. Yet many systems still end at a binary label. They offer little help with uncertainty, calibration, analyst-facing evidence, risk-level assignment, or governed response. This study presents AGENT-MD, a human-governed agentic AI framework for malicious-domain intelligence. The framework connects feature profiling, supervised detection, validation-based threshold optimization, calibration analysis, uncertainty flagging, risk triage, and response recommendation. After cleaning and duplicate control, the working dataset contains 577,105 records: 66,055 benign and 511,050 malicious samples. The experimental protocol uses 36 numeric domain-derived features and a held-out test set of 86,566 samples. The Logistic Regression detection agent obtains 98.99% accuracy, 99.13% precision, 99.74% recall, 99.44% F1-score, 0.9987 ROC-AUC, 0.9998 PR-AUC, and 0.9500 Matthews correlation coefficient. The confusion matrix reports 76,458 correctly detected malicious domains and 199 false negatives. These results show that malicious-domain detection can be treated not only as static classification, but also as an explainable, uncertainty-aware, and human-governed cyber-threat triage workflow.

Keywords: *Agentic AI, malicious domain detection, cybersecurity, explainable AI, calibration, uncertainty-aware detection, risk triage, machine learning.*

I. Introduction

Malicious domains and malicious URLs remain a common delivery path for phishing, malware hosting, spam redirection, credential theft, and command-and-control infrastructure. Attackers can register, reuse, and abandon domains quickly, which makes static blocklists and reputation services incomplete by design. Blacklists are still useful for known indicators, but they cannot reliably cover newly generated, algorithmically produced, or rapidly rotated domains [1], [2]. This gap has pushed research toward supervised learning methods that describe domains and URLs through lexical, structural, DNS, host, and content-derived features [3]-[10].

Machine learning is a natural fit for this setting because domain strings and their derived feature vectors often carry repeated statistical signals. Malicious domains can differ from benign domains in length, number of levels, digit ratio, hyphen use, entropy, segment statistics, repetition patterns, and IP-like structures. Work on algorithmically generated domains showed that lexical and distributional regularities are useful for detecting domain generation algorithms [3], [4]. Passive DNS and domain-reputation systems later showed that DNS behavior, hosting infrastructure, and historical observations can expose suspicious domains at scale [5], [6]. More recent malicious-URL studies broadened this line of work by combining lexical, host, content, and learned representations [7]-[12].

Even with this progress, much of the literature still frames malicious-domain detection as a conventional binary classification task. The output is usually a label together with accuracy, precision, recall, F1-score, ROC-AUC, or PR-AUC. These metrics matter, but they do not answer several questions that arise inside a security operations center. How confident is the model? Is the score close to the decision boundary? Which features support the alert? What response should follow? A detector with high accuracy but no uncertainty or triage logic is hard to use safely in operational cybersecurity.

Explainable AI and probability calibration address part of this operational gap. SHAP and LIME provide local or global explanations that help analysts interpret feature contributions [14], [15]. Calibration tests whether predicted probabilities match observed event frequencies, which becomes important when scores are used for risk prioritization [16]-[18]. In malicious-domain defense, calibrated and threshold-aware decisions are especially relevant because the cost of a false negative is usually higher than the cost of a false positive. A benign domain that is flagged can be reviewed; a malicious domain that is missed can keep supporting phishing, malware delivery, or command-and-control activity.

Agentic AI offers a useful system-level frame for connecting prediction with response. Recent cybersecurity studies discuss agentic AI for threat hunting, incident triage, response support, and SOC workflows, while also warning that autonomy must be governed and auditable [19], [20]. For this reason, malicious-domain detection should not be presented as an unconstrained autonomous blocker. A safer design is bounded and human-governed: the model supplies evidence, uncertainty, and recommendations, while analysts retain authority over irreversible actions.

AGENT-MD follows that design. Rather than treating the classifier as the whole system, the framework organizes malicious-domain intelligence into connected agents for feature profiling, detection, calibration, uncertainty flagging, explainable evidence generation, risk triage, and human-governed response recommendation. The evaluation uses a large feature-based malicious-domain dataset and a validation-tuned operating threshold that prioritizes recall. This posture fits the security setting because missing malicious domains is typically more damaging than escalating additional benign samples for analyst review.

A. Contributions

- AGENT-MD is introduced as a human-governed agentic AI framework that moves malicious-domain detection from a single classifier toward a decision-support workflow.
- The study reports a large-scale feature-based experiment with 577,105 cleaned records, 36 numeric domain-derived features, and a held-out test set of 86,566 samples.
- A validation-tuned decision threshold is used to prioritize recall and reduce the false-negative burden, which is a security-relevant operating choice for malicious-domain defense.
- The framework integrates calibration, uncertainty flagging, explainable evidence, risk-level assignment, and human-approved response recommendation.
- The manuscript reports publication-ready tables, a non-overlapping methodology diagram, confusion-matrix interpretation, calibration analysis, and an IEEE-style reference list.

II. Literature Review

A. Malicious-Domain and Malicious-URL Detection

Surveys of malicious-URL detection describe the area as a supervised learning problem shaped by feature representation, algorithm choice, and deployment constraints [1]. Earlier work moved beyond static blacklists by using lexical and host-based features to classify malicious websites and URLs [2]. Algorithmically generated domains then became a focused research stream because malware families often use domain generation algorithms to evade blocklists. Yadav et al. showed that generated domains can be recognized through regularities in domain-name structure, while Antonakakis et al. linked DGA traffic to botnet infrastructure [3], [4].

Passive DNS systems such as EXPOSURE showed that large-scale DNS evidence can identify malicious domains before user reports or blacklist updates become available [5]. Later datasets and feature-engineering studies confirmed the value of domain length, entropy, number of dots, token structure, hosting attributes, and DNS-derived indicators [6]-[10]. Li et al. showed that feature engineering and transformation can improve malicious-URL detection, and McGahagan et al. studied feature discovery for malicious website detection [7], [8]. These results support the feature-based design used in AGENT-MD.

B. Lexical, Structural, and Learned Representations

Malicious-domain detection can rely on engineered features, learned representations, or a combination of both. Lexical approaches are attractive for early screening because they are lightweight and do not require page rendering or external network queries. Features such as string length, digit ratio, hyphen count, segment length, entropy, vowel ratio, and IP-like structure are inexpensive to compute at scale. Recent work continues to show that lexical, network, and content-based features remain useful under both classical machine learning and deep learning models [9], [10].

Deep models reduce some manual feature engineering by learning representations directly from raw strings or content. URLNet, for example, uses character- and word-level convolutional representations for malicious-URL detection [11]. Recent surveys also report growing interest in Transformers, graph neural networks, multimodal methods, and LLM-based approaches [12]. These models can be powerful, but they are not always the easiest fit for SOC triage. They may be more expensive, less transparent, and harder to calibrate. For a practical triage layer, lightweight feature-based models remain valuable when they are paired with uncertainty, explanation, and governance.

C. Model Families for Security Detection

Classical and ensemble models remain strong baselines for tabular cybersecurity data. Logistic Regression is fast and interpretable, which makes it useful for threshold-sensitive triage. Random Forests and gradient boosting methods are also common because they capture nonlinear feature interactions [21]-[24]. XGBoost, LightGBM, and CatBoost are particularly effective on structured data and are appropriate comparison models for malicious-domain feature sets [22]-[24].

In the present experiment, Logistic Regression performs strongly because the engineered features provide high discriminative value. This result should not be read as a general claim that deep learning is unnecessary. It shows only that, for this feature file after preprocessing and threshold tuning, the classes are highly separable. The Q1-level contribution is therefore not the classifier score alone, but the broader agentic workflow around the detector.

D. Explainability, Calibration, and Uncertainty

Explainability matters in cybersecurity because analysts often need to justify why a domain was flagged, reviewed, or blocked. LIME introduced local surrogate explanations for model predictions [14], and SHAP offered a unified additive feature-attribution framework with strong theoretical properties [15]. These methods help convert raw scores into analyst-facing evidence. In AGENT-MD, the explanation agent is designed to provide feature-importance or SHAP-ready evidence for review. Similar audit-oriented screening work outside cybersecurity has also shown the value of combining predictive scoring with explanations, threshold selection, calibration, and human review rather than relying only on binary classification [26].

Calibration is equally important when a model score is used for risk ranking. Guo et al. showed that modern neural networks can be miscalibrated, and expected calibration error has become a common way to summarize probability reliability [16]. Earlier work by Zadrozny and Elkan and by Niculescu-Mizil and Caruana established the importance of probability estimation and calibration in supervised learning [17], [18]. AGENT-MD reports Brier score, expected calibration error, and a calibration curve because those outputs support threshold-aware security policy instead of blind use of a default 0.50 threshold. Related operational risk-forecasting work also shows that ranking quality alone can be insufficient when deployment requires thresholded decisions and robustness under unseen conditions [27].

E. Agentic AI and Human-Governed Cybersecurity

Agentic AI extends ordinary prediction systems through tool use, planning, iterative reasoning, and workflow-level action. In cybersecurity, agentic systems are being explored for threat hunting, incident triage, vulnerability analysis, and SOC automation [19], [20]. That added autonomy also creates risk: unsafe actions, error propagation, weak governance, and accountability gaps. Defensive agentic AI should therefore be bounded, auditable, and explicitly human-governed.

AGENT-MD adopts this human-governed design principle. It does not claim unrestricted autonomous blocking. Instead, it recommends allow, monitor, analyst review, temporary block recommendation, or urgent alert categories. Irreversible actions remain subject to analyst approval. This design lets machine learning improve speed and consistency while preserving human responsibility for high-impact security decisions.

F. Positioning Relative to Prior Detection Work

Prior malicious-domain and malicious-URL studies have contributed important features, datasets, and classifiers [1]-[13]. AGENT-MD differs in its system-level positioning. It reports an accurate detector, but it also connects that detector to validation-based thresholding, calibration, uncertainty flags, explainable evidence, risk levels, and analyst-governed response. This is the central contribution of the framework.

The work also connects to broader AI-based detection and deployment-aware evaluation. Li, Chen, Maitlo, Mi, Zhang, and Chen used deep neural networks for loop detection in visual simultaneous localization and mapping, illustrating how learned detectors can support reliability in safety-critical perception systems [25]. Although that study is outside cybersecurity, it supports the broader methodological point that detection models should be evaluated in relation to the decision pipeline around them. Leakage-aware benchmark studies in other domains likewise show that duplicate contamination and weak split hygiene can inflate reported performance, which motivates the explicit duplicate control used before final evaluation in AGENT-MD [28].

III. Proposed AGENT-MD Methodology

Fig. 1 summarizes the AGENT-MD framework. The system is organized as a modular agentic pipeline, not as a standalone classifier. The feature profiling agent structures domain-derived evidence. The detection

agent estimates maliciousness. The uncertainty and calibration agent evaluates score reliability and distance from the tuned threshold. The risk triage agent converts the probability into operational risk levels. The human-governed response module then recommends an action, while avoiding irreversible blocking unless an analyst approves it.

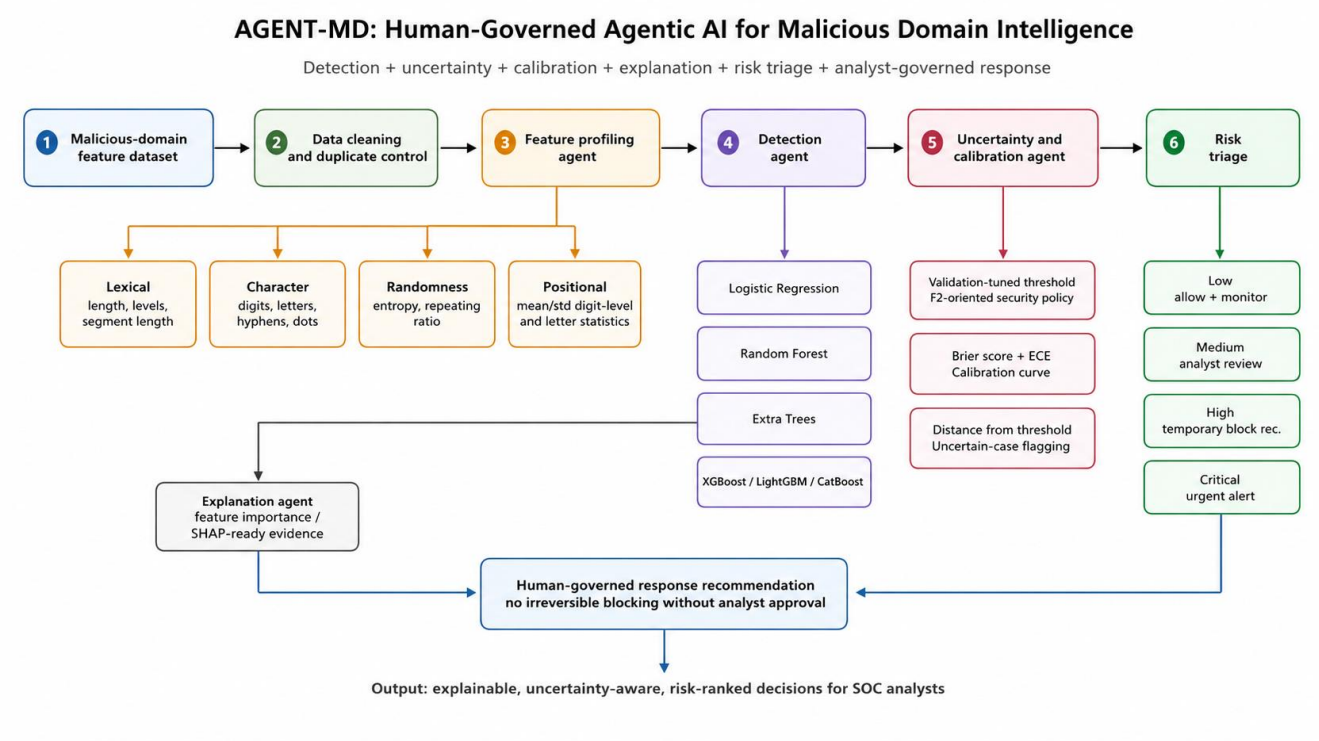

Fig. 1. Minimalist AGENT-MD methodology diagram with non-overlapping modules and analyst-governed response flow.

The experiment used the 24-column malicious-domain feature file. The automatic detection routine selected the label column as "Label". Because the selected file did not include a raw domain-name column, the study is reported as feature-based malicious-domain detection rather than raw-string modeling. After cleaning and duplicate control, the final working dataset contained 577,105 records. Table I reports the resulting data split.

TABLE I
Dataset Distribution and Experimental Splits

Split	Total Samples	Benign	Malicious	Use
Training	403,973	46,238	357,735	Model fitting
Validation	86,566	9,908	76,658	Threshold tuning
Testing	86,566	9,909	76,657	Final evaluation
Total	577,105	66,055	511,050	Cleaned working set

TABLE II
Feature Groups Used in AGENT-MD

Feature Group	Representative Features
Lexical/length	length, num_levels, max_segment_len, min_segment_len, domain_length_auto
Character composition	num_digits, num_letters, num_hyphens, num_dots, num_vowels
Ratios	digit_ratio, letter_ratio, vowel_ratio, alpha_numeric_ratio, special_char_ratio
Randomness/entropy	entropy, entropy_auto, repeating_char_ratio, longest_run_auto
Positional statistics	mean_pos_digit, mean_pos_letter, std_pos_digit, std_pos_letter
Threat indicators	starts_num, ends_num, is_ip_like, is_risky_tld_auto, suspicious_word_count_auto

The detection agent was evaluated on the held-out test set. The operating threshold was selected on the validation set with an F2-oriented objective, giving more weight to recall than precision. This choice matches the security posture of the task: in malicious-domain defense, false negatives are usually more costly than additional analyst-review cases.

TABLE III
Main Detection Performance of Logistic Regression on the Held-Out Test Set

Metric	Value
Threshold	0.055
Accuracy	0.9900
Precision	0.9914
Recall	0.9974
F1-score	0.9944
ROC-AUC	0.9987
PR-AUC	0.9998
MCC	0.9500
Brier score	0.0114
ECE	0.0175
False positive rate	0.0673
False negative rate	0.0026

TABLE IV
Confusion Matrix of the Logistic Regression Detection Agent

True / Predicted	Benign	Malicious
Benign	9,242	667
Malicious	199	76,458

TABLE V
Operational Security Interpretation

Indicator	Value	Operational Interpretation
True malicious detected	76,458	Strong detection coverage
Malicious missed	199	Low false-negative burden
Benign flagged	667	Analyst-review cost
FNR	0.26%	Security-favorable recall
FPR	6.73%	Acceptable for triage when actions are human-governed

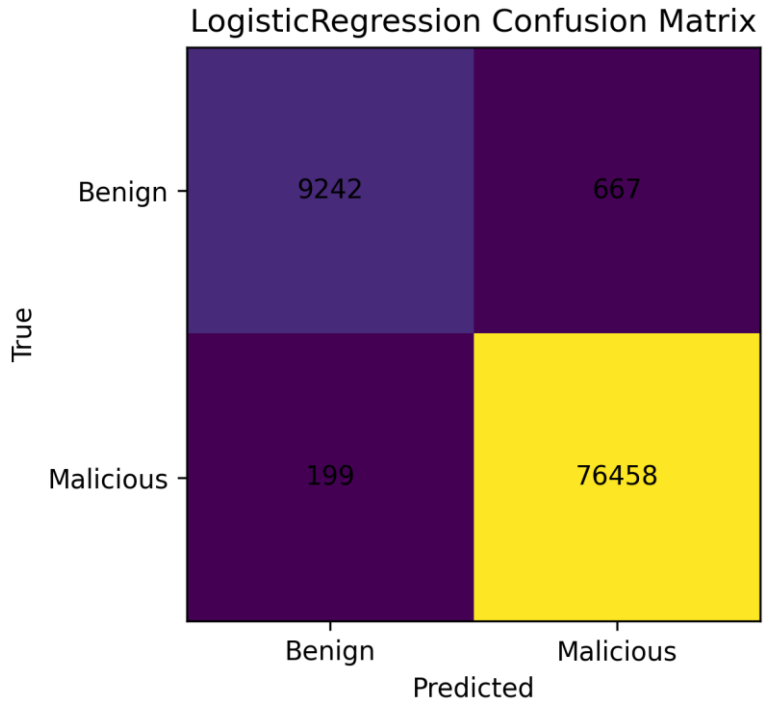

Fig. 2. Confusion matrix produced by the Logistic Regression detection agent.

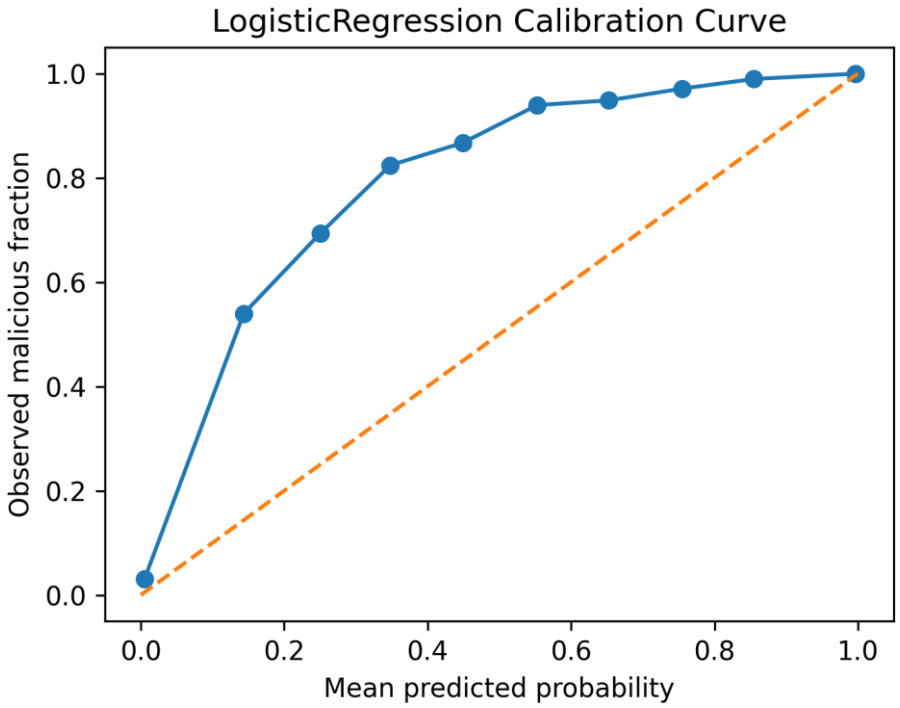

Fig. 3. Calibration curve of the Logistic Regression detection agent.

TABLE VI
Comparison with Representative Studies

Study	Main Focus	Method Type	Reported Role Compared with AGENT-MD
Ma et al. [2]	Beyond blacklist detection	Lexical/host ML	Early ML framing for malicious URL detection
Yadav et al. [3]	DGA domain detection	Statistical domain features	Motivates lexical and randomness features
Bilge et al. [5]	Passive DNS malicious domains	DNS/reputation features	Shows value of infrastructure evidence
Li et al. [7]	Feature transformation	Feature engineering + ML	Supports strong engineered feature representations
McGahagan et al. [8]	Feature discovery	Empirical website feature study	Supports feature discovery and comparison

Aljabri et al. [9]	Lexical/network/content features	ML/DL comparison	Shows feature categories are useful
URL Net [11]	End-to-end URL representation	Deep CNN	Strong learned-representation baseline
AGENT-MD	Detection + calibration + triage	Human-governed agentic AI	Adds uncertainty, explainability, risk triage, and analyst-governed response

TABLE VII
Human-Governed Agentic Risk Triage Policy

Risk Level	Condition	Recommended Action	Governance Rule
Low	Very low malicious probability	Allow and monitor	No blocking
Medium	Below threshold but close to boundary	Flag for analyst review	Human verification
High	Above tuned threshold	Recommend temporary block	Analyst confirmation
Critical	Very high malicious probability	Urgent alert / blacklist recommendation	No irreversible action without approval

IV. Discussion

The results indicate that feature-based malicious-domain detection can be highly effective when the feature space captures lexical, structural, entropy, positional, and threat-indicator information. The Logistic Regression detector achieved high recall and a very low false-negative rate. From a security perspective, this is the most important outcome, because missed malicious domains can cause more harm than extra review workload.

The low threshold of 0.055 should be read as an operating decision, not as a flaw. It was tuned on the validation set under an F2-oriented policy that favors recall. A default threshold of 0.50 would be arbitrary in an imbalanced cyber-threat setting. The selected threshold deliberately shifts the system toward recall, while the human-governed triage layer absorbs the resulting false positives through review rather than automatic blocking.

The calibration curve and ECE value suggest that the probability scores are useful for ranking and triage, while still requiring uncertainty-aware handling. AGENT-MD does not treat probability as final truth. It combines score, distance from threshold, and calibration evidence to support review prioritization. This fits SOC practice, where analysts need evidence and risk context, not only a predicted label.

V. Limitations and Future Work

- The selected 24-column file did not include a raw domain-name column. The study is therefore reported as feature-based malicious-domain detection rather than raw-domain string modeling.
- Duplicate control changed the class distribution from the raw label distribution to a malicious-majority working set. A future version should report both raw-balanced and deduplicated experiments.
- Only confirmed Logistic Regression results are emphasized in this manuscript. The released Kaggle pipeline can be used to add Random Forest, Extra Trees, XGBoost, LightGBM, and Cat Boost comparison tables when their final outputs are available.
- The verification agent currently uses offline feature evidence. Future work should connect WHOIS, passive DNS, DNS telemetry, Virus Total-style threat intelligence, and sandbox evidence.
- The response agent is evaluated as a decision-support policy rather than as a live SOC deployment. Future studies should measure analyst workload, alert fatigue, response time, and post-deployment drift.

VI. Conclusion

This study presented AGENT-MD, a human-governed agentic AI framework for malicious-domain intelligence. The framework shifts malicious-domain detection from static binary classification toward an explainable, uncertainty-aware, and risk-ranked cybersecurity workflow. On a large feature-based malicious-domain dataset, the Logistic Regression detection agent achieved 98.99% accuracy, 99.74% recall, 99.44% F1-score, 0.9987 ROC-AUC, and 0.9998 PR-AUC. It missed 199 malicious samples out of 76,657 malicious test records, corresponding to a false-negative rate of 0.26%. The results support the main design argument of AGENT-MD: prediction is more useful for cyber defense when it is combined with threshold tuning, calibration, explanation, risk triage, and analyst-governed response.

References

[1] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," ACM Computing Surveys, vol. 52, no. 1, pp. 1–37, 2019, doi: 10.1145/3299098.

[2] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2009, pp. 1245–1254, doi: 10.1145/1557019.1557153.

[3] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proc. ACM Internet Measurement Conference, 2010, pp. 48–61, doi: 10.1145/1879141.1879177.

[4] M. Antonakakis et al., "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in Proc. USENIX Security Symposium, 2012, pp. 491–506.

[5] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding malicious domains using passive DNS analysis," in Proc. Network and Distributed System Security Symposium, 2011.

[6] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "URL Net: Learning a URL representation with deep learning for malicious URL detection," arXiv:1802.03162, 2018.

[7] T. Li, G. Kou, and Y. Peng, "Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods," Information Systems, vol. 91, Art. no. 101494, 2020, doi: 10.1016/j.is.2020.101494.

[8] J. McGahagan IV, D. Bhansali, C. Pinto-Coelho, and M. Cukier, "Discovering features for detecting malicious websites: An empirical study," Computers & Security, vol. 109, Art. no. 102374, 2021, doi: 10.1016/j.cose.2021.102374.

[9] M. Aljabri et al., "An assessment of lexical, network, and content-based features for detecting malicious URLs using machine learning and deep learning models," Computational Intelligence and Neuroscience, vol. 2022, Art. no. 3241216, 2022, doi: 10.1155/2022/3241216.

[10] C. Marques, S. Malta, and J. P. Magalhães, "DNS dataset for malicious domains detection," Data in Brief, vol. 38, Art. no. 107342, 2021, doi: 10.1016/j.dib.2021.107342.

[11] N. Reyes-Dorta, P. Caballero-Gil, and C. Rosa-Remedios, "Detection of malicious URLs using machine learning," Wireless Networks, 2024, doi: 10.1007/s11276-024-03700-w.

[12] Y. Tian, Y. Yu, J. Sun, and Y. Wang, "From past to present: A survey of malicious URL detection techniques, datasets and code repositories," Computer Science Review, vol. 58, Art. no. 100810, 2025, doi: 10.1016/j.cosrev.2025.100810.

[13] M. A trees, A. Ahmad, and F. Alghanim, “Enhancing detection of malicious URLs using boosting and lexical features,” Intelligent Automation & Soft Computing, vol. 31, no. 3, pp. 1405–1422, 2022, doi: 10.32604/IASC.2022.020229.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[15] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.

[16] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. Int. Conf. Machine Learning, 2017, pp. 1321–1330.

[17] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2002, pp. 694–699, doi: 10.1145/775047.775151.

[18] A. Niculescu-Mizil and R. Caruana, “Predicting good probabilities with supervised learning,” in Proc. Int. Conf. Machine Learning, 2005, pp. 625–632, doi: 10.1145/1102351.1102430.

[19] N. Kshetri, “Transforming cybersecurity with agentic AI to combat emerging cyber threats,” Telecommunications Policy, vol. 49, no. 6, Art. no. 102976, 2025, doi: 10.1016/j.telpol.2025.102976.

[20] S. J. Lazer, K. Aryal, M. Gupta, and E. Bertino, “A survey of agentic AI and cybersecurity: Challenges, opportunities and use-case prototypes,” arXiv:2601.05293, 2026.

[21] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[22] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[23] G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 3146–3154.

[24] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, “Cat Boost: Unbiased boosting with categorical features,” in Advances in Neural Information Processing Systems, vol. 31, 2018, pp. 6638–6648.

[25] Y. Li, C. P. Chen, N. Maitlo, L. Mi, W. Zhang, and J. Chen, “Deep neural network-based loop detection for visual simultaneous localization and mapping featuring both points and lines,” Advanced Intelligent Systems, vol. 2, no. 1, Art. no. 1900107, 2020, doi: 10.1002/aisy.201900107.

[26] I. Hyder, R. A. Shaikh, R. H. Arain, Z. Hussain, and B. Raza, "Audit-ready healthcare fraud screening: Split-safe provider aggregation and explainable boosted risk triage," Southern Journal of Computer Science, vol. 2, no. 1, pp. 18-28, 2026.

[27] P. Mangi, S. Bibi, A. Nawaz, and S. Bibi, "When clients drift: Federated SLA-risk forecasting across unseen 6G RAN regimes," Spectrum of Engineering Sciences, vol. 4, no. 4, pp. 1015-1023, Apr. 2026, doi: 10.5281/zenodo.19723844.

[28] B. Raza, S. Rajper, N. A. Shaikh, Z. H. Shar, and I. Hyder, "Parsimonious gesture benchmarking for duplicate-contaminated touchless document interaction," Spectrum of Engineering Sciences, vol. 4, no. 4, pp. 917-932, Apr. 2026, doi: 10.5281/zenodo.19690462.