



IJRTSM

INTERNATIONAL JOURNAL OF RECENT TECHNOLOGY SCIENCE & MANAGEMENT

“COGCYBER: A COGNITIVE CYBER THREAT INTELLIGENCE FRAMEWORK USING BIG DATA ANALYTICS AND MACHINE LEARNING FOR REAL-TIME THREAT DETECTION AND AUTOMATED RESPONSE”

Kundan Kumar Mishra¹, Dr. Amaravathi Pentaganti²

^{1,2} Department of Computer Science and Engineering, NIILM University, Kaithal, Haryana, India

kundankumarmishra@gmail.com

ABSTRACT

Cyber threat intelligence (CTI) enables organizations to anticipate, detect and respond to attacks, yet the volume, velocity and variety of the data it depends on overwhelm conventional, manual analysis. This paper presents CogCyber, a cognitive CTI framework that combines big-data technologies (Apache Kafka, Hadoop HDFS and Spark), open-source CTI platforms (MISP and OpenCTI), machine-learning classifiers and a security orchestration, automation and response (SOAR) layer to collect, enrich, analyze and act on structured and unstructured threat data from social media, open-source intelligence (OSINT) feeds, network and system logs, and security tools. The architecture covers data acquisition, pre-processing, feature extraction, modeling and visualization. CogCyber was implemented as a Security Operations Center (SOC) testbed and evaluated with simulated attacks and a fraudulent-e-mail (phishing) detection use case. Across 20–50 input features, a Random Forest classifier achieved approximately 0.97–0.98 accuracy, precision, recall and F1-score, compared with approximately 0.92–0.95 for Naïve Bayes, and the SOAR layer triggered automated quarantine, blocking and notification actions without analyst intervention.

Key Words: Cyber Threat Intelligence, Big Data Analytics, Machine Learning, Random Forest, Naïve Bayes, SOAR, Phishing Detection, Security Operations Center.

I. INTRODUCTION

Cyber-attacks are growing in frequency, sophistication and destructiveness. Every additional connected device and system widens the attack surface, and adversaries continually refine their tactics, for example by exploiting zero-day vulnerabilities and by persuading users to execute infected files. Perimeter controls such as firewalls and signature-based antivirus tools are reactive by design and cannot, on their own, keep pace.

Cyber threat intelligence (CTI) is the process of acquiring, analyzing and disseminating information about potential or actual threats to an organization's assets. Good intelligence describes the tactics, techniques and procedures (TTPs) of threat actors and the indicators of compromise (IoCs) that can be used to detect and mitigate their attacks, and it must be relevant, timely, accurate and actionable. [1], [2] The difficulty is that the data feeding CTI is generated by modern information systems and networks at a volume, velocity and variety that traditional processing techniques cannot handle. [3], [4]

Big-data technologies such as Hadoop, Spark and NoSQL stores now make it feasible to store and process such data in near real time, and machine-learning and natural-language-processing (NLP) techniques can extract threat knowledge from it. [5], [6] Existing proposals, however, tend to address one stage of the CTI life cycle (collection, analysis or sharing) or to be limited to particular threat types, and few connect detection to an automated response.

<https://www.ijrtsm.com> © International Journal of Recent Technology Science & Management

This paper proposes **CogCyber**, a cognitive CTI framework that closes this loop. The main contributions are:

- An end-to-end architecture that ingests structured and unstructured data from social media, OSINT feeds, network and system logs, SIEM, IDS/IPS, firewalls, malware-detection and forensic tools into a distributed big-data platform.
- A pipeline that enriches processed data with IoC and TTP context from MISP and OpenCTI and classifies it with machine-learning models for real-time detection.
- Integration of the detection output with a SOAR layer (Cortex, TheHive, Splunk Phantom) for automated containment and analyst notification.
- An experimental evaluation on a SOC testbed, comparing Naïve Bayes (NB) and Random Forest (RF) classifiers over a range of feature-set sizes.

The remainder of the paper is organized as follows. Section II reviews related work, Section III describes the methodology, Section IV presents the CogCyber framework, Section V reports the results, Section VI discusses challenges and limitations, and Sections VII and VIII give future directions and conclusions.

II. LITERATURE REVIEW

Research on CTI automation has progressed along four lines: learning-based analysis, big-data platforms, knowledge extraction from text, and intelligence sharing.

A. Learning-Based and Data-Fusion Approaches

Aryalet *et al.* combine data fusion and deep learning: data from the dark web, honeypots and social media are cleaned, fused into a unified view of the threat landscape using source dependability, timeliness and relevancy, and then classified. [7] Sinha *et al.* mine data from several sources and classify cyber-threat indicators with a modified Naïve Bayes algorithm, reporting better precision and accuracy than earlier classifiers. [8] Cho *et al.* argue that explainability improves the utility of machine-learning-based CTI and propose a process of feature selection, training, assessment and enhancement that integrates it. [9]

B. Big-Data Platforms for CTI

CTI-CORE uses big-data analytics over network traffic, archives and threat-intelligence inputs, organized as acquisition, pre-processing and analysis components. [10] Liu *et al.* and Narayanan *et al.* describe Hadoop, Spark, Elasticsearch and Kafka-based pipelines and evaluate them on precision, recall and accuracy for a malware case study. [5], [11] Zhang *et al.* integrate SOAR with CTI on big-data technologies to improve incident response, and surveys by Bucur and Sgârciu, Chen *et al.* and Zhang *et al.* catalogue data sources, machine-learning strategies and open challenges such as data quality, processing capacity and privacy. [3], [4], [6], [12], [13]

C. Knowledge Extraction and Sharing

Vulcan applies NLP to unstructured text (news, blogs and social media) to extract entities, events and threat actors, [14] and earlier systems mine open-source reports for IoCs automatically or compute over provenance data to find threats. [15], [16] SECDFAN extends collection to discussion forums. [17] K-CTIAA places CTI from news, social media and dark-web forums in a knowledge graph to expose relationships among threat entities. [18] cyberaCTive builds on the STIX format to bring CTI into complex system models. [19] A family of formats and protocols (STIX, TAXII, OpenIOC, CyBOX, IODEF, MISP, CIF and CRITs) standardizes how intelligence is described and exchanged, [1], [2] and surveys of sharing platforms point to technical compatibility, legal and regulatory compliance, trust and privacy as the main barriers. [20], [21] Lin *et al.* additionally correlate CTI with sightings to assess and augment intelligence. [22] Table 1 summarizes representative work and its limitations.

TABLE 1 Representative CTI Approaches and Their Limitations

Work	Technique	Limitation
Aryal et al.	Deep learning with data fusion	Limited to specific threat types (malware, phishing, DDoS)
Cho et al.	Explainable machine learning	Specific ML algorithms and use cases
Jo et al. (Vulcan)	NLP on unstructured text	Limited to threats against software
Arul et al.	Review of CTI sharing platforms	Limited to existing platforms
Alsaleh et al.	Systematic review of CTI sharing	Limited to published literature
CogCyber (this work)	Big data + CTI enrichment + ML + SOAR	Evaluated on one feature dataset and a testbed (Sec. VI)

The gap that CogCyber addresses is the absence of a single, deployable pipeline that spans collection, enrichment, machine-learning detection and automated response, evaluated in operation rather than only described.

III. RESEARCH METHODOLOGY: CYBER-RESILIENCE FRAMEWORK

The study follows an experimental design science approach: a framework is specified, implemented as a working SOC, and evaluated against simulated attacks and a labeled dataset using standard classification and operational metrics.

A. Experimental Testbed

The testbed is a SOC built from open-source components. Big-data processing uses Apache Kafka, Hadoop HDFS and Spark. CTI is handled by MISP and OpenCTI. The SOAR component is assembled from Cortex, TheHive, Elasticsearch and Kibana, with Splunk Phantom used for playbook-driven automated responses. Machine-learning models are trained on the enriched data and applied in real time.

B. Data Sources

The platform ingests ten classes of data (Table 2). External sources provide intelligence on emerging threats, while internal sources provide telemetry from the protected environment. Model training and evaluation use real-world network, system and security-log style data and a labeled phishing feature set described in Section V.

TABLE 2
Data Sources Ingested by CogCyber

Source	Role in the framework
Social media	Early indicators of emerging threats
OSINT feeds	Publicly available intelligence
Network traffic logs	Detection of suspicious flows
System event logs	Security-relevant host activity
SIEM	Correlated security events and alarms
IDS / IPS	Detected intrusions and attack patterns
Firewalls	Blocked, unauthorized or anomalous traffic
Malware-detection units	Malware trends and detection rates
Forensics tools	Evidence and attack-vector context

C. Evaluation Metrics

Classifier quality is measured with accuracy, precision, recall and F1-score, computed from true/false positives and negatives (TP, FP, TN, FN):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

At the operational level the effectiveness of the CTI program is judged by detection time, false-positive rate, the number of incidents successfully resolved, threat coverage, the amount of actionable intelligence produced, response time and cost-effectiveness. A high false-positive rate, in particular, floods analysts with false alarms and wastes resources.

IV. THE COGCYBER FRAMEWORK

CogCyber is organized as a six-stage pipeline (Fig. 1), in which each stage feeds the next and the response stage feeds back into enrichment and model updates.

A. Data Acquisition and Ingestion

Apache Kafka acquires streams from social media, OSINT feeds, network traffic, system and security logs. Its high-throughput, fault-tolerant design allows sources to be added without redesign. Ingested data are persisted in HDFS for later batch analysis.

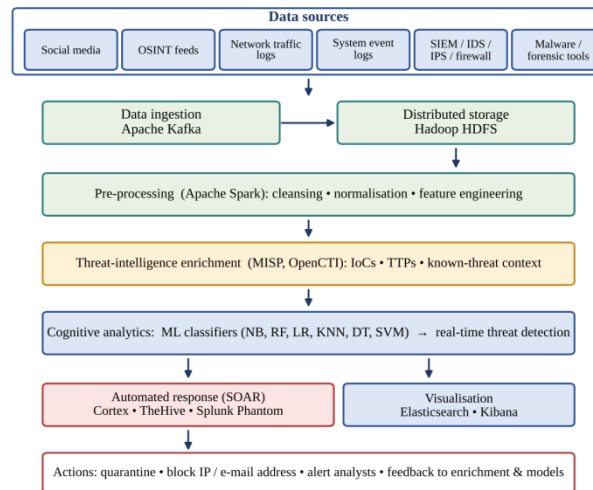


Fig. 1. *The CogCyber architecture: from multi-source data acquisition to automated response.*

B. Pre-processing

Apache Spark performs data cleansing, normalization and feature engineering in parallel across the cluster. The cleaned data are written back to HDFS, which keeps the real-time path and the historical path consistent.

C. Threat-Intelligence Enrichment

Processed records are correlated with external intelligence through MISP and OpenCTI. The enrichment attaches context about known threats, vulnerabilities and IoCs (malicious IP addresses, domains, URLs and file hashes) to each record, so that classification works on data that already carries the latest intelligence.

D. Cognitive Analytics

Machine-learning classifiers are trained on the enriched data to identify and categorize threats. The framework supports logistic regression, k-nearest neighbors, decision trees, Random Forest, support vector machines and Naïve Bayes; this paper reports the evaluation of NB and RF. Trained models are applied to live streams for real-time detection, and their outputs are forwarded to the response layer.

E. Automated Response and Visualization

Detection results are consumed by the SOAR component (Cortex, TheHive and Splunk Phantom). Playbooks quarantine affected systems or messages, block malicious IP or e-mail addresses and alert security personnel. Elasticsearch and Kibana provide dashboards so that analysts can inspect detections, trends and the outcome of automated actions.

V. EXPERIMENTAL ANALYSIS AND RESULTS

A. Setup and Use Case

The CogCyber testbed was exercised with a sequence of simulated attacks. Threat intelligence was loaded into MISP and OpenCTI, and rules were defined to trigger automated responses in Splunk Phantom and TheHive through Cortex. As a representative use case, the framework was tasked with detecting and responding to a fraudulent (phishing) e-mail: archives were ingested through Kafka, cleansed, normalized and feature-engineered with Spark, stored in HDFS, enriched with MISP/OpenCTI context and then used to train NB and RF models. The trained models classified fraudulent messages in real time and passed their verdicts to the SOAR layer, which quarantined the messages, blocked the sender addresses and notified security staff. Models were evaluated on feature subsets of 20 to 50 features.

B. Feature Analysis

Machine-learning detection of malicious URLs and new phishing forms is well established, [23], [24] and the feature groups used here cover host-based, URL-based, sub-domain, URL-form and page-content characteristics. Fig. 2 shows the correlation of one such group with the class label. Across the five feature groups examined, the strongest correlations with the label were observed for the percentage of external null/self-redirect hyperlinks (-0.54), frequent domain-name mismatch (0.46), the number of dashes in the URL (-0.37), submission of information to e-mail (-0.36), null/self-redirect hyperlinks (0.34) and insecure forms (0.32). Most individual features were only weakly correlated with the label, which motivates the use of models that combine many features.

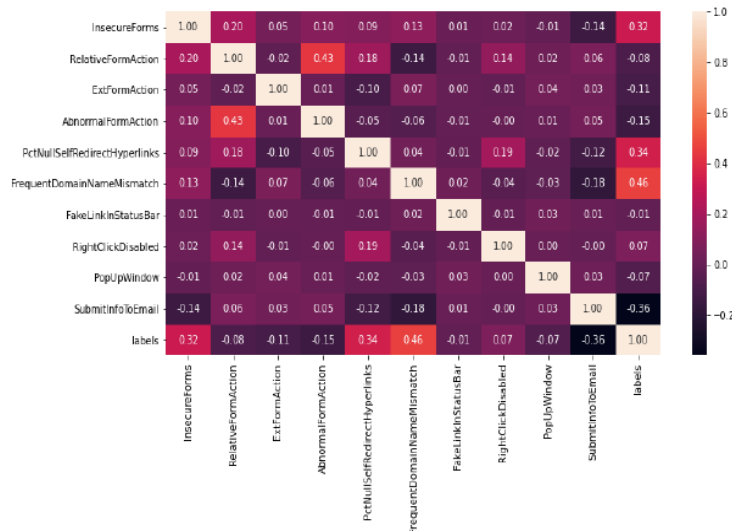


Fig. 2. Correlation heatmap of URL-form features with the class label (the labels row/column).

C. Classifier Performance

Fig. 3 and Fig. 4 plot precision, recall, F1-score and accuracy of the NB and RF models as the number of features increases from 20 to 50. Table 3 summarizes the approximate ranges read from these plots. NB shows a clear upward trend with feature count, with its weakest precision (about 0.90) at 27 features and its best results at 45–50 features. RF is both more accurate and far less sensitive to the feature count, remaining within a narrow band of approximately 0.97–0.98 for accuracy and F1-score.

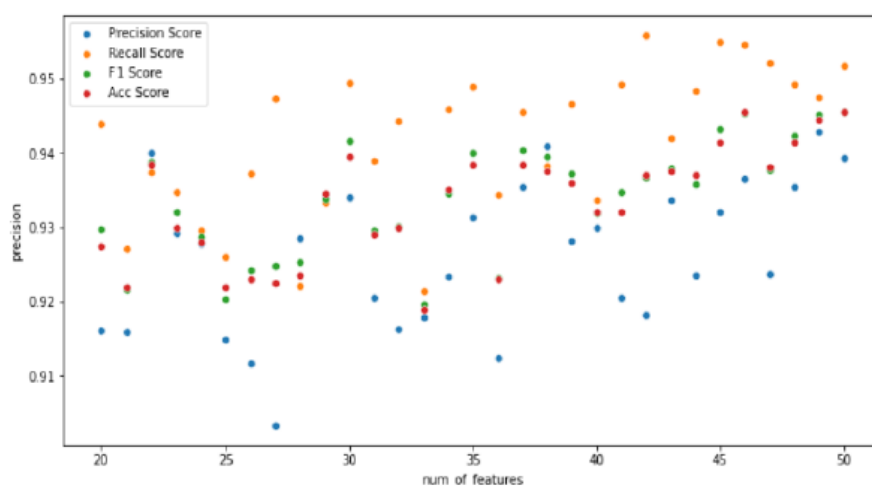


Fig. 3. Precision, recall, F1-score and accuracy of the Naïve Bayes model versus the number of features.

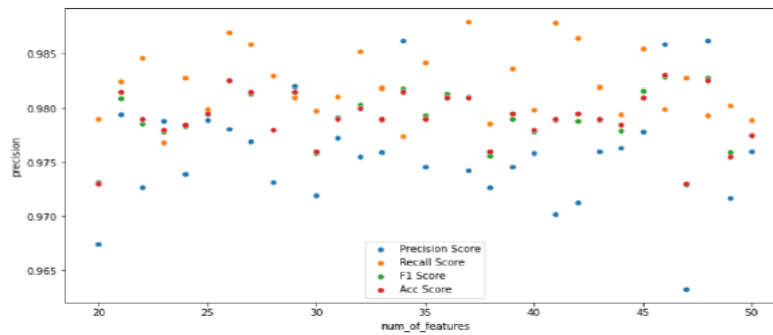


Fig. 4. Precision, recall, F1-score and accuracy of the Random Forest model versus the number of features.

TABLE 3 Approximate Classifier Performance (20–50 Features)

Metric	NB range	NB at 50	RF range	RF at 50
Accuracy	0.92–0.95	≈0.945	0.97–0.98	≈0.977
Precision	0.90–0.94	≈0.939	0.96–0.99	≈0.976
Recall	0.92–0.96	≈0.952	0.97–0.99	≈0.979
F1-score	0.92–0.95	≈0.945	0.97–0.98	≈0.977

At 50 features the RF model leads NB by roughly three percentage points in accuracy and F1-score and by about 3.7 points in precision. The practical consequence is fewer false alarms for analysts and fewer missed threats, which are the two operational concerns highlighted by the CTI metrics in Section III.

D. Automated Response

In the simulated attacks and in the phishing use case, the end-to-end chain of ingestion, enrichment, classification and SOAR execution operated in real time. Detected threats caused quarantine, blocking and notification actions to be executed automatically through Splunk Phantom and TheHive. The integration of CTI and big-data technologies made it possible to collect, process and analyze large quantities of data from many sources and to act on the result promptly.

E. Discussion

The results support three observations. First, enrichment and classification are complementary: CTI context supplies what is known, whereas the learned model generalizes beyond known indicators. Second, an ensemble method such as RF is a better default than NB for heterogeneous, partly correlated features, because NB assumes feature independence while many URL and page features are visibly correlated (Fig. 2). Third, connecting detection to SOAR shortens the interval between detection and containment, which is where most damage occurs.

VI. CHALLENGES AND LIMITATIONS

A. Technical Challenges

Volume and velocity demand scalable storage and stream processing, and data quality varies considerably between sources. Explainability of machine-learning verdicts is needed for analyst trust. [9] Adversaries may also craft inputs that evade learned detectors, and deep-learning detectors for malware remain costly to train and interpret. [25]

B. Organizational and Legal Challenges

Legal and regulatory obstacles, data-privacy concerns and a lack of standardization make it hard to share CTI between organizations. [20], [21] Operating a SOC of this kind also requires skilled staff and sustained investment.

C. Limitations of This Study

The classifier evaluation uses a single labeled phishing feature dataset, and the results in Table 3 are ranges read from scatter plots across feature subsets, not repeated cross-validated runs; confidence intervals are not reported. The use case concerns fraudulent e-mails and web pages, so performance on other threat classes (for example malware or intrusion traffic) remains to be measured. The attacks were simulated in a testbed rather than observed in a production network.

VII. FUTURE SCOPE

Future work will (i) extend evaluation to further datasets and threat classes with cross-validation and statistical testing; (ii) add advanced feature engineering and data-mining techniques to raise model accuracy; (iii) introduce explainable models and knowledge-graph reasoning; (iv) adopt standardized sharing protocols and technologies such as blockchain for trustworthy intelligence exchange; and (v) add proactive threat hunting on top of the existing pipeline.

VIII. CONCLUSION

This paper presented CogCyber, a cognitive CTI framework that unites big-data ingestion and processing, CTI enrichment, machine-learning detection and SOAR-based response. In a SOC testbed, the Random Forest classifier delivered approximately 0.97–0.98 accuracy, precision, recall and F1-score across 20–50 features, clearly outperforming Naïve Bayes (approximately 0.92–0.95), and the SOAR layer automated containment. The framework offers organizations a practical route from raw, heterogeneous threat data to timely, actionable intelligence, and its modular design allows the sources, models and playbooks to evolve with the threat landscape.

REFERENCES

- [1] V. Mavroeidis and S. Bromander, "Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence," arXiv:2103.03530, 2021.
- [2] A. Ramsdale, S. Shiales, and N. Kolokotronis, "A comparative analysis of cyber threat intelligence sources, formats and languages," *Electronics*, vol. 9, no. 5, p. 824, 2020.
- [3] D. Bucur and V. Sgârciu, "Big data analytics and machine learning for cyber threat intelligence: A survey," *Sensors*, vol. 21, no. 15, 2021.
- [4] Y. Zhang, Y. Qian, J. Zhang, and L. Sun, "Big data analytics for cyber threat intelligence: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 118883–118896, 2020.
- [5] S. Liu, H. Li, J. Li, and L. Sun, "Big data analytics for cyber threat intelligence: A survey," *IEEE Access*, vol. 9, pp. 123018–123034, 2021.
- [6] W. Chen, Q. Wu, C. Zhao, Q. Wu, and Y. Liu, "A big data approach to cyber threat intelligence analytics," *IEEE Access*, vol. 8, pp. 162023–162038, 2020.
- [7] R. Aryal, Y. M. Jang, S. Lee, and J. Kim, "A novel cyber threat intelligence framework based on deep learning and data fusion," *J. Inf. Process. Syst.*, 2020.
- [8] A. Sinha, V. Yadav, and N. Yadav, "An approach to improve cyber threat intelligence using data mining techniques," *Int. J. Netw. Secur.*, vol. 22, no. 2, pp. 216–225, 2020.
- [9] K. Cho, J. Shin, and Y. Kim, "Enhancing cyber threat intelligence with explainable machine learning," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2047–2061, 2021.
- [10] I. Alkinj, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "CTI-CORE: A cyber threat intelligence framework for big data analysis," *IEEE Access*, vol. 9, pp. 98404–98415, 2021.
- [11] S. Narayanan, K. Chandrasekaran, and V. Ganapathy, "A big data framework for cyber threat intelligence analytics," *Int. J. Comput. Appl.*, vol. 181, no. 26, pp. 9–15, 2018.
- [12] L. Zhang, X. Yao, S. Liu, S. Liu, and H. Li, "A SOAR and CTI framework with big data technologies," *IEEE Access*, vol. 8, pp. 151379–151388, 2020.
- [13] I. Kotenko, I. Saenko, and A. Brantskiy, "Machine learning and big data processing for cybersecurity data analysis," in *Data Science in Cybersecurity and Cyberthreat Intelligence*. Springer, 2020, pp. 61–85.

- [14] H. Jo, Y. Lee, and S. Shin, "Vulcan: Automatic extraction and analysis of cyber threat intelligence from unstructured text," *Comput. Secur.*, vol. 120, p. 102763, 2022.
- [15] X. Liao *et al.*, "Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 755–766.
- [16] X. Shu *et al.*, "Threat intelligence computing," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 1883–1898.
- [17] G. Sakellariou, P. Fouliras, and I. Mavridis, "SECDFAN: A cyber threat intelligence system for discussion forums utilization," *Eng.*, vol. 4, no. 1, pp. 615–634, 2023.
- [18] Z.-X. Li *et al.*, "K-CTIAA: Automatic analysis of cyber threat intelligence based on a knowledge graph," *Symmetry*, vol. 15, no. 2, p. 337, 2023.
- [19] R. M. Czekster, R. Metere, and C. Morisset, "cyberaCTive: A STIX-based tool for cyber threat intelligence in complex models," arXiv:2204.03676, 2022.
- [20] R. Arul, K. Karthikeyan, and K. A. Chandrasekaran, "Comprehensive analysis of cyber threat intelligence sharing platforms," *J. Inf. Secur. Appl.*, vol. 62, 2021.
- [21] M. Alsaleh, R. Alturki, and F. Alhomoud, "Cyber threat intelligence sharing: A systematic review and future directions," *IEEE Access*, vol. 9, pp. 45212–45227, 2021.
- [22] P.-C. Lin *et al.*, "Correlation of cyber threat intelligence with sightings for intelligence assessment and augmentation," *Comput. Netw.*, vol. 228, 2023.
- [23] B. Sun, T. Takahashi, L. Zhu, and T. Mori, "Discovering malicious URLs using machine learning techniques," in *Data Science in Cybersecurity and Cyberthreat Intelligence*. Springer, 2020, pp. 33–60.
- [24] C. N. Gutierrez *et al.*, "Learning from the ones that got away: Detecting new forms of phishing attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 988–1001, 2018.
- [25] M. Gopinath and S. C. Sethuraman, "A comprehensive survey on deep learning based malware detection techniques," *Comput. Sci. Rev.*, vol. 47, 2023.