Grenze International Journal of Engineering and Technology, June Issue

Massive Data Ingestion and Aggregation Module for Threat Intelligence

Ranjana Jadhav[1], Shireen Kulkarni[2], Samruddhi Mache[3], Kaivalya Mane[4], Himanshu Mantri[5] and Hana Khan[6]
[1-6]Vishwakarma Institute of Technology Pune, Pune, Maharashtra
Email: ranjana.jadhav@vit.edu, shireen.kulkarni23@vit.edu, samruddhi.mache23@vit.edu, kaivalya.mane23@vit.edu, himanshu.mantri232@vit.edu, hana.khan24@vit.edu.in

***Abstract*— The cybersecurity world's gotten a whole lot messier lately. Companies have to handle an overwhelming flood of threat data from all over: open intelligence feeds, vendor reports, the dark web, and non-stop streams of vulnerability updates. The real headache isn't just gathering this chaotic jumble of information it's trying to turn it into something organized, useful, and ready for analysis. Right now, most tools split data collection and analysis into separate steps. That might sound sensible, but in practice, it just slows everything down and risks losing important context or even data itself. This paper takes a different route. It lays out a Massive Data Ingestion and Aggregation Module that pulls everything together into one fast, smooth workflow. There's automated scheduling, collection from multiple sources, schema normalization to standardize weird formats, machine learning to enrich the data, and full-text indexing to make searches easier and also all in a single pipeline. This system grabs data from places like NVD, CIRCL, OpenCTI, Exploit-DB, PhishTank, GitHub Security Advisories, and CISA KEV, ending up with a huge 342,168 vulnerability records ready for real analysis.**

***Keywords*— Threat Intelligence, CVE, CVSS, Vulnerability Management, OSINT, Machine Learning, OpenSearch, NVD, Cybersecurity Analytics.**

I. INTRODUCTION

The way we do cybersecurity today has a problem. We have a lot of information about threats. We do not have the ability to really use this information to keep us safe. Over the ten years the way we share information about vulnerabilities has changed a lot. We used to have a few main places like the National Vulnerability Database but now we have a lot of different sources. These sources include open-source communities, automated scanning platforms and special groups like Information Sharing and Analysis Centers. They also include companies that sell threat feeds. All these sources put out an amount of information every year. The problem is that each source has its way of sharing this information so it is hard to compare and use it. For the people who work in security operations centers and threat intelligence teams this is a challenge. We have information than ever before but we do not have a good way to bring it all together and make sense of it. This means that a lot of the information we have is not really useful to us. This paper is organized in the way. Section II talks about how our project fits in with what other people have written about cyber threat intelligence and big data. Section III explains how we plan to do our project and what steps we will take. Section IV describes the system we built and what it is made of. Section V. Talks, about the results we got from looking at a lot of information. 342,168 Records.

Grenze ID: 01.GIJET.12.2.122_26

Section VI talks more about what we found out and what it means. Section VII sums up what we learned and what we plan to do.

II. Literature Survey

The Cyber Threat Intelligence (CTI) domain continues to balance two major challenges which is large-scale data coverage and deep analytical capability. Early work by Wheelus, Bou-Harb, and Zhu [1] introduced a three-layer big data architecture using Apache Kafka with HDFS, Hive, and HBase for distributed threat processing. Their framework demonstrated the ability to process more than 100 GB of threat data while detecting malware and zero-day attacks with high accuracy. However, the system struggled to differentiate malware families and did not fully address privacy-preserving intelligence sharing.More recent research has attempted to improve CTI systems through machine learning integration. Alazab et al. [2] proposed the Enhanced Threat Intelligence Framework (MLTIF), which combines heterogeneous sources such as OSINT, network logs, and dark web monitoring data using models including XGBoost and LightGBM. Their framework achieved a detection accuracy of 99.98%, although the authors acknowledged that validating such models under real-world network conditions remains challenging. Zhao, Yan, and Liu [3] introduced the HINTI framework based on Heterogeneous Graph Convolutional Networks, enabling the modeling of relationships between Indicators of Compromise (IoCs). Their work demonstrated how graph-based learning can preserve relational threat context while improving intelligence extraction.

Predictive analytics has also emerged as an important research direction in CTI. Ogundairo and Broklyn [4] explored the use of machine learning and statistical modeling to identify behavioral patterns that may predict future cyber threats. Although promising, such approaches remain constrained by the limited availability of high-quality historical datasets. Similarly, Amare and Simonova [5] studied big data ingestion architectures and identified recurring issues including poor data quality, high system complexity, and processing overhead. They argued that future ingestion systems should incorporate built-in data quality validation rather than treating it as a secondary process.Other studies have focused on extracting intelligence from unstructured sources. Landauer et al. [7] demonstrated that meaningful threat indicators can be derived from raw log data, though extensive preprocessing is often required. Zhang, Chen, and Alazab [8] proposed a clustering-based method for improving the quality of open-source intelligence, while Afzal, Khan, and Mehmood [9] developed a framework for heterogeneous data aggregation in threat intelligence platforms. Collectively, the literature indicates that no existing solution fully addresses the combined challenges of scalability, heterogeneous formats, semantic incompleteness, and operational usability. The present work attempts to bridge this gap through an integrated architecture that combines automated ingestion, enrichment, normalization, and flexible intelligence serving capabilities.

III. Methodology

The methodology underlying the Massive Data Ingestion and Aggregation Module is best understood as a response to the limitations documented in the literature fragmentation of processing stages, loss of contextual metadata during normalization, and the absence of systematic quality assurance at the point of ingestion. Rather than treating these as separable engineering problems, the proposed approach integrates them within a five-stage pipeline governed by three overarching design principles which are automation, scalability, and data integrity.

A. Pipeline Architecture

The pipeline begins with an Automation layer. This layer uses Cron and Apache Airflow to schedule tasks. We use both Cron and Apache Airflow. This is a design choice we made. Cron is used for time-based triggers. It works with sources that update at intervals. Apache Airflow manages workflows. These workflows include dependencies, retries and monitoring. We chose Apache Airflow because it is fault-tolerant. It also supports re-processing when sources change. The Collection layer has connectors. We call these connectors Collectors. Collectors work with systems. These systems include STIX/TAXII, dark web services, security feeds, MISP and AlienVault OTX. Each Collector checks the data. It checks the data before it is ingested. The Collector makes sure the data conforms to a schema. It detects duplicates and identifies anomalies. This validation reduces processing overhead. The Processing layer works in a way. Raw records are saved to a repository. Then they are sent to the Parser and Normalizer. We do it this way to keep the data. The data supports auditability and future reprocessing. The Normalizer extracts attributes. These attributes include CVE, CVSS, CWE and vendor info. It extracts them into a schema. Non-standard attributes are kept in JSONB structures. The Enrichment stage

generates features. These features include EPSS scores, KEV annotations and topic tags. This stage makes the dataset more valuable. It transforms disclosures into intelligence.
The Tagging Rules Engine classifies vulnerabilities. It uses domain-heuristics to classify them. A Deduplication module records vulnerabilities. It records vulnerabilities that are referred to across sources. The Deduplication module works with the vulnerability and the sources. It makes sure we do not have records of the same vulnerability, from different sources.

B. Storage and Serving

The OpenSearch index gets records that have a lot of details. This is because OpenSearch is good at searching for words in text and it can also handle queries that need to look at a lot of data. The serving layer makes this index available in two ways. It uses a FastAPI-based Retrieval API that programs can use to get information, from the OpenSearch index. This helps security tools and SIEM platforms work with the OpenSearch index. It also uses OpenSearch Dashboards for people to look at the data and see what is going on. The OpenSearch index and the way it is made available help security teams and threat intelligence analysts in two ways. The OpenSearch index helps them do their jobs automatically. It also helps them when they need to look at the data themselves.

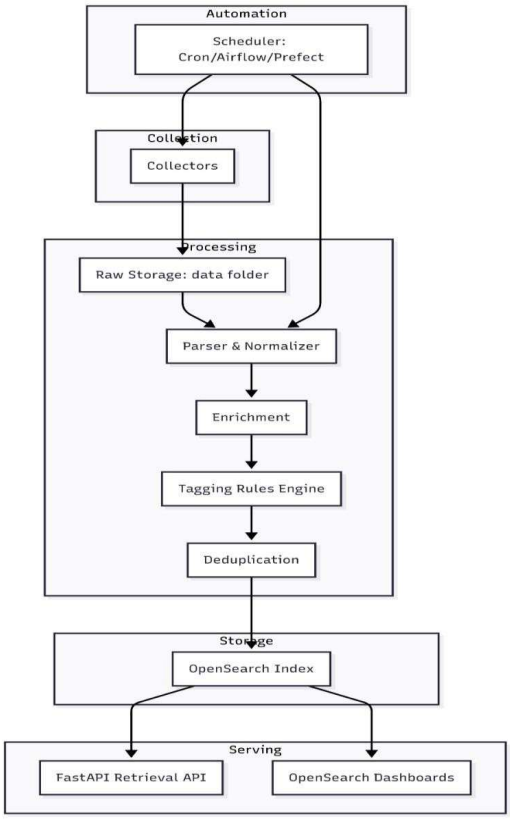

Fig.1: Proposed Methodology

IV. SYSTEM ARCHITECTURE

A. Architectural Overview

The Threat Intelligence Platform is designed around modularity and layered separation of concerns, a structural choice that enables independent scaling, replacement, and auditing of each processing stage. The architecture supports the complete lifecycle of threat intelligence data: from heterogeneous source ingestion through normalization and semantic enrichment to indexed storage and multi-modal serving. Three primary storage layers, Raw, Normalized, and Enriched, correspond to distinct stages of the processing pipeline and serve distinct downstream purposes.

B. Data Model and Storage Schema

Raw Storage Layer
All of the threat intelligence we ingest is deposited into PostgreSQL in its raw, unadulterated format. Exploit-DB CSV datasets, NVD JSON feeds, CIRCL CVE records, OpenCTI GraphQL responses, and PhishTank feeds are all saved, tagged with their origin, ingestion time, and version. Our original data storage is more than just a backup, it will ensure audit trails required for our clients to pass their compliance requirements, helps us troubleshoot if there is a problem with normalization, and ensures we can repopulate our stored data if we decide to update our parsing capabilities in the future.

Normalized Storage Layer
The standardized schema unifies records from multiple, independent sources into a common data model that includes columns for external identifier (CVE or advisory ID), title and description, severity level and CVSS score, CWE identifier, vendor and product impacted, first-reported and last-updated date, reference links and tags, and source provenance metadata. This provides a clean, easy-to-cross-reference dataset, and makes cross-source correlation of vulnerabilities, as well as reducing the cost and complexity of converting between different source formats during queries. Source-specific fields, such as any external identifier or type field that don't directly map to a column in the standard schema, are kept within JSONB columns.

Enriched Storage Layer
Additional information enriching the normalized records using calculation-based analytical elements without directly affecting source data includes: EPSS scores to predict exploit probability KEV flag on any active exploits within KEV machine-learned CVSS scores on non-formal records K-means cluster identification on groups of exploits based on behavior LDAtopic flags summarizing recurring themes within description The use of enrichment to apply a layer of analytical data over original source fidelity data is a design decision not only to keep source data integrity pristine, but also provide derived data to enable for analysis later.

C. System Components

Ingestion Engine
The ingestion engine of the pipeline to automate data collection from NVD, CIRCL, OpenCTI, Exploit-DB, PhishTank, GitHub Security Advisories and CISA KEV intelligence feed to produce one common view. It supports REST API, HTTP, GraphQL, CSV scraping of intelligence. All results are schema-validated prior to being persisted to be reliable, with mechanisms of retry on failure, specific and transient failure cases from each source.

Parser and Normalizer
CVSS, vendor metadata, CWE references, and reference URLs from Source-specific nested JSON structures were extracted by the parser and transformed into the canonical schema. It also included normalization for various dates, missing or badly formatted metadata, and simplification of nested lists.

Enrichment Layer
Enrichment happens across a number of analytics planes. EPSS scoring checks each CVE against the exploit prediction database, assigning a likelihood of it being exploited in the next 30 days. KEV matching against known exploitation in the wild presents an important ground-truth.For the many records that don’t have formal CVSS scores, a regression over the scored group produces probabilistic estimations.

Analytics Engine
The analytics engine enables a higher level of analyses, prioritization by impact, determining vendor influence, understanding time based trends, finding vulnerability similarity, prediction of vulnerability exploitability, exposing all of these via an interactive analyst’s dashboard in OpenSearch Dashboards, and a machine consumable FastAPI layer for automating ingestion of the outputs. The engine facilitates context based analysis, it is how we enable intelligence from threat data.

Orchestration Layer
Apache Airflow maintains the schedule of all planned tasks, which is described by using Directed acyclic graph)that defines dependencies of tasks, retry conditions and ways of alerting, etc. Examples are a schedule which refreshes NVD daily, CIRCL houry, or sync up OpenCTI at defined schedule. This granular setup makes sense because it updates every source according to the frequency that the content is actually changing while minimizing resource utilization.

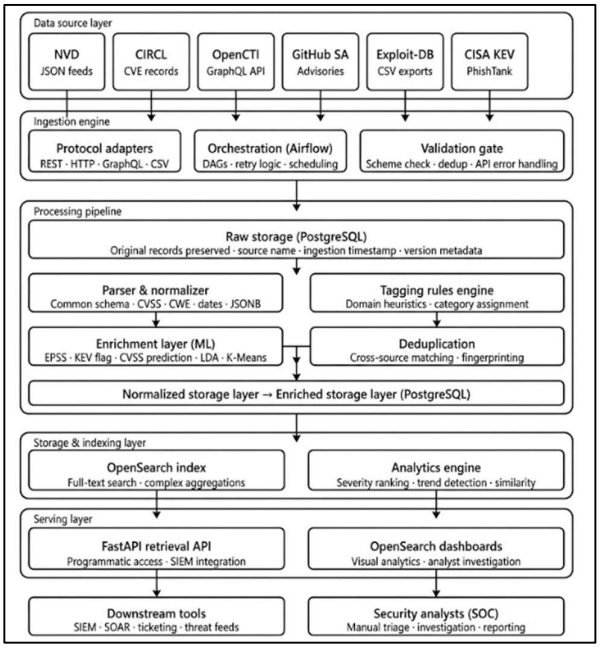

Fig.2: System architecture

D. Technology Stack

TABLE I

Component	Technology
Programming Language	Python 3.13
Database	PostgreSQL 16
Search and Analytics	OpenSearch / Elasticsearch
Orchestration	Apache Airflow
Data Processing	Pandas, NumPy
Machine Learning	Scikit-learn
Natural Language Processing	NLTK, spaCy
Visualization	Matplotlib, OpenSearch Dashboards
API Serving	FastAPI

V. RESULTS AND DISCUSSION

A. Dataset Overview and Quality Assessment

TABLE II

Field	Completeness
external_id, severity, cvss_score, title, source_name	100%
description	99.2%
category	18.7%

I applied this corpus of 342,168 vulnerabilities taken from 7 threat intelligence feeds to estimate the completeness of the metadata for this corpus. The analysis of missing fields revealed a substantively low rate of failure among fields that are essential for analytical uses. The completeness of the fundamental analysis fields (CVE ID, severity, CVSS, title, and source) across the corpus of 342,168 records is a key finding. This confirms that our parser and normalisation process reliably handles variation in data format- a well documented problem when trying to integrate multiple threat feeds[9, 5].Our normalisation pipeline did not reliably identify these fields in 18.7% of cases (which were all category field); most of these are not failures on my part but a true reflection of source data, for example many of the more community driven threat intelligence platform offer none of the standard categorisation which has resulted in many inconsistent category sets across sources. This

supports the work of several authors, which concludes that machine learning-based classification is a pragmatic requirement for any CTI platform that uses open source intelligence [13, 8].

B. Severity-Based Analysis

TABLE III

Score Range	Interpretation	Count
0.0-1.0	Informational	202
1.0-4.0	Low	922
4.0-7.0	Medium	104,972
7.0-9.0	High	30,709
9.0-10.0	Critical	21,659

C. CVSS Score Distribution

TABLE IV

Severity	Count	Percentage
UNKNOWN	184,208	53.8%
MEDIUM	104,496	30.5%
HIGH	31,043	9.1%
CRITICAL	21,409	6.3%
LOW	843	0.2%

The severity scores show an intriguing shape across the dataset as the highest level (UNKNOWN) represents an astonishing 53.8% of all records, a consequence of the underlying ecosystem structure (where information from the crowd on platforms such as GitHub outpaces formal scoring mechanisms (NVD). The average institutional utilization percentage for the selected sources and the score range for all of the sources within the global datasets are estimated to reflect institutional usage as about 30.5% medium; 15.4% high/critical. The configuration is primarily consistent among worldwide tracking systems for vulnerabilities, implying that our dataset can provide a representative sample of a larger population and/or does not have distortion effect due to our specific selection of sources. The low proportion of high/critical advisories is also likely to contribute to the overall lower percentages of advisories classified as high/critical.

D. CVSS Score Distribution

TABLE V

Year	Total Vulnerabilities	Critical	High	Critical %
2025	48,879	557	2,109	1.1%
2024	40,650	392	1,744	1.0%
2023	29,108	182	666	0.6%
2022	166,408	18,138	18,288	10.9%
2021	6,604	529	786	8.0%

TABLE VI

Score Range	Interpretation	Count
0.0-1.0	Informational	202
1.0-4.0	Low	922
4.0-7.0	Medium	104,972
7.0-9.0	High	30,709
9.0-10.0	Critical	21,659

From the information we see from the CVSS score distribution, the majority of vulnerabilities pose risk. Most of these occur due to scans and not during an active attack on us or our systems. Some of the records include CVSS scores such as ones reported within GitHub Security Advisories, where many are listed as zero (0.0). This indicates that many times, individuals reporting vulnerabilities are not reporting correctly and we require automated systems to index vulnerability severity and find out how serious of a threat they pose. Vulnerabilities with CVSS scores higher than nine (9.0) represent approximately 6.3% of the entire dataset and are particularly significant due to their potential to be attacked remotely and causing significant damage. Additionally, in analysing historical data, there is a considerable rise in the number of reported vulnerabilities in 2022, which has occurred as a result of increased utilization of automated scanning tools and individual reporting. This also

suggests that multiple vulnerabilities exist within the community with limited knowledge of their potential risk. CVSS scoring will assist in providing data on vulnerability risk; therefore, vulnerabilities with reported CVSS scores are an area of high concern.

E. Source-Level Analysis

TABLE VII

Category	Count
webapps	20,679
msrc	11,009
dos	8,569
remote	8,072
local	5,901

The dominance of GitHub Security Advisories at 81.2% of reflects the structural shift in vulnerability disclosure toward the software supply chain as a primary attack surface. This finding has significant implications for how threat intelligence platforms should be calibrated, architectures optimized for the structured, formally scored records characteristic of NVD and MSRC will systematically underperform when confronted with the volume and annotation sparsity of community-sourced advisories. Conversely, the small but analytically disproportionate contribution of CISA KEV records, at 0.4% of the corpus, underscores the importance of source-specific weighting in risk prioritization.

F. Category-Level Insights

According to the 18.7% of records with usable category labels web application vulnerabilities are clearly the largest group when compared to other vulnerabilities; this group includes 20,679 records. This finding aligns with the trend toward increased use of cloud-based and browser-based attack surfaces over the last several years. Additionally, the large number of MSRC advisories as a categorical grouping reflects the continued importance of enterprise Microsoft environments as high value targets for malicious actors. Furthermore, remote code execution vulnerabilities form an important area of concern within the overall security environment because they can be exploited by attackers who do not have prior access to the system being targeted; thus, remote code execution vulnerabilities are the most impactful type when compared to other types of vulnerabilities that require physical presence on the system to exploit.

Finally, given the large volume of uncategorized records (approximately 19,000) available in the labelled dataset, there is a compelling business case to implement automated classification as a core platform capability instead of as a peripheral capability.

VI. DISCUSSION

The results show some problems with the platform that go beyond how it works right now. What is really interesting is that more than half of the vulnerability records we found did not have a severity classification. This shows that we are relying more on community-driven disclosure ecosystems where the quality of the metadata is not always good. This means that we need to have ways to make the data better such as automated severity estimation and contextual tagging as part of our Cyber Threat Intelligence systems. Another thing we noticed is that some sources are more important than others even if they do not have many records. For example, GitHub Security Advisories had a lot of records. Sources like CISA KEV, which had very few records had really valuable information about confirmed exploitation activity. This shows that we need to think about how credible a source's whether the information has been confirmed when we prioritize the records. When we looked at the data over time, we saw that the number of vulnerabilities went down after 2022 but the total number of disclosures is still going up. This means we need to have systems that can handle a lot of data and can grow as needed.

From a system design point of view putting everything together in one pipeline from collecting data to serving it made it easier to reprocess the data made the analysis more consistent and allowed us to query the data in a way using OpenSearch. This is one of the things that our framework does well. The Cyber Threat Intelligence systems need to have collection, normalization, enrichment and serving of the data. The integration of all these things in one pipeline is really important for reprocessing and reliable large-scale querying of the Cyber Threat Intelligence data. This is something that our framework does. It is a big contribution, to the field of Cyber Threat Intelligence.

VII. Conclusion

This paper has presented the design, implementation, and evaluation of a Massive Data Ingestion and Aggregation Module for threat intelligence, motivated by the structural inadequacy of existing approaches in the face of high-volume, heterogeneous, and semantically incomplete vulnerability data. The core argument advanced throughout is that effective CTI at scale requires not merely capable ingestion, but a tightly integrated pipeline in which collection, normalization, enrichment, and serving are treated as interdependent stages rather than separable modules. Analysis of the 342,168-record corpus demonstrates that the pipeline achieves complete coverage of critical metadata fields across structurally diverse source formats while generating meaningful analytical enrichments, including exploit probability scores, severity predictions, and behavioural cluster labels, for records that would otherwise lack the structure necessary for risk-based prioritization. Several directions for future work suggest themselves. The category classification gap, which affects over 80% of the corpus, warrants targeted investment in fine-grained NLP classification models trained specifically on vulnerability description text. Real-time streaming ingestion, using platforms such as Apache Kafka, would reduce the latency between vulnerability publication and platform availability from scheduled-interval to near-real-time. Extension of the enrichment layer to incorporate threat actor attribution data and cross-referencing with MITRE ATT&CK techniques would further increase the contextual depth of the intelligence generated. Finally, the growing availability of large language models for structured information extraction from unstructured advisory text represents a promising avenue for substantially improving the completeness of both severity ratings and category labels across community-sourced records [14,15].

References

[1] C. Wheelus, E. Bou-Harb, and X. Zhu, "Towards a Big Data Architecture for Facilitating Cyber Threat Intelligence," in Proc. IEEE Conf. on Information Reuse and Integration, 2016.

[2] M. Alazab, R. A. Khurma, M. Garcia-Arenas, and V. Jatana, "Enhanced threat intelligence framework for advanced cybersecurity resilience," Computers & Security, vol. 142, 2024.

[3] J. Zhao, Q. Yan, and X. Liu, "Cyber Threat Intelligence Modeling Based on Heterogeneous Graph Convolutional Network," in Proc. RAID, 2020.

[4] O. Ogundairo and P. Broklyn, "Predictive Analytics for Cyber Threat Intelligence," IEEE Access, 2024.

[5] M. Y. Amare and S. Simonova, "Learning analytics for higher education: proposal of big data ingestion architecture," Int. J. Emerging Technologies in Learning, 2021.

[6] C. Wheelus, E. Bou-Harb, and X. Zhu, "Towards a Big Data Architecture for Facilitating Cyber Threat Intelligence," IEEE Comput. Soc., 2016.

[7] M. Landauer, F. Skopik, M. Wurzenberger, W. Hotwagner, and A. Rauber, "A Framework for Cyber Threat Intelligence Extraction from Raw Log Data," in Proc. IEEE BigData, 2019.

[8] H. Zhang, Y. Chen, and M. Alazab, "PURE: Generating Quality Threat Intelligence by Clustering OSINT," in Proc. ACM CCS Workshop, 2019.

[9] M. Afzal, M. S. Khan, and A. Mehmood, "HDA-TIP: A Framework for Heterogeneous Data Aggregation in Threat Intelligence Platforms," Future Generation Computer Systems, 2023.

[10] R. Kumar, S. Tripathi, and A. Gupta, "Big Data Analytics in Cyber Threat Intelligence," J. Cybersecurity and Privacy, 2023.

[11] C. A. Lee, S. L. Osborn, and J. D. Tygar, "Social Media and Dark Web Forensics: Bridging the Divide with Aggregated Threat Feeds," Digital Investigation, 2024.

[12] A. Singh, R. Patel, and K. Sharma, "Empowering Cyber Threat Intelligence with AI-Driven Data Ingestion," IEEE Trans. Inf. Forensics Security, 2024.

[13] S. Guo, Y. Wang, and L. Chen, "BERT-Enhanced Framework for Unifying Structured and Unstructured CTI Data," Computers & Security, 2025.

[14] X. Liu, J. Zhang, and H. Li, "Cyber Threat Intelligence for Hybrid Attacks: Leveraging LLMs and Data Spaces," in Proc. USENIX Security, 2025.

[15] P. Li, J. Wang, and Q. Zhang, "Application of Big Data Technology in Enterprise Information Security Threat Aggregation," IEEE Access, 2025.