МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ/METHODS AND SYSTEMS OF INFORMATION PROTECTION, INFORMATION SECURITY

DOI: https://doi.org/10.60797/IRJ.2026.163.57

MFCTIIF: MULTI-FEED CYBER THREAT INTELLIGENCE INTEGRATION FRAMEWORK

Research article

Iasenovets A.V.[1,] *, Tang F.[2]
[1] ORCID : 0009-0008-0008-3746;
[2] ORCID : 0000-0002-0048-9876;
[1, 2] Chongqing University of Posts and Telecommunications, Chongqing, China

* Corresponding author (archee.busy[at]gmail.com)

Abstract

The growing volume and diversity of cyber threat intelligence (CTI) feeds pose significant challenges to interoperability, metadata consistency, and automated threat assessment. In this paper, we present MFCTIIF — Multi-Feed Cyber Threat Intelligence Integration Framework designed to aggregate, enrich, and classify malware indicators from heterogeneous sources. The system performs structured data matching, synonym resolution, and automated threat-level classification, outputting JSON-formatted feeds suitable for security operations. We evaluate MFCTIIF on the latest 100 malware samples revealing the majority of them represent high-risk threats such as RATs and consistent end-to-end processing latency of ≈15–17s per sample. A comparative analysis against four existing frameworks demonstrates that MFCTIIF is the only system to fulfill all seven key attributes, however, it is constrained by metadata gaps and classification imbalance for unknown threats. To address this, we propose future enhancements including automated mapping to STIX by LLM models, LLM-driven classification, fuzzy matching, and parallelized caching to improve coverage and latency.

Keywords: cyber threat intelligence, threat classification, multi-source data integration, malware analysis, framework.

MFCTIIF: ИНТЕГРАЦИОННАЯ ПЛАТФОРМА ДЛЯ МНОГОКАНАЛЬНОЙ ОБРАБОТКИ ДАННЫХ О КИБЕРУГРОЗАХ

Научная статья

Ясеновец А.В.[1,] *, Тан Ф.[2]
[1] ORCID : 0009-0008-0008-3746;
[2] ORCID : 0000-0002-0048-9876;
[1, 2] Чунцинский университет почты и телекоммуникаций, Чунцин, Китай

* Корреспондирующий автор (archee.busy[at]gmail.com)

Аннотация

Растущий объем и разнообразие потоков данных разведки киберугроз (CTI) создают серьезные проблемы для взаимодействия, согласованности метаданных и автоматизированной оценки угроз. В этой статье мы представляем MFCTIIF — интеграционную платформу многоканальной разведки киберугроз, предназначенную для агрегации, обогащения и классификации индикаторов вредоносных программ из разнородных источников. Система выполняет структурированное сопоставление данных, разрешение синонимов и автоматизированную классификацию уровня угроз, выводя потоки в формате JSON, подходящие для операций по обеспечению безопасности. Мы тестируем MFCTIIF на основе 100 последних образцов вредоносных программ, показывая, что большинство из них представляют собой угрозы высокого риска, такие как RAT, и имеют постоянную задержку сквозной обработки ≈15–17 с на образец. Сравнительный анализ с четырьмя существующими решениями показывает, что MFCTIIF — единственная система, отвечающая всем семи ключевым требованиям, однако она ограничена пробелами в метаданных и дисбалансом классификации для неизвестных угроз. Для решения этой проблемы мы предлагаем будущие усовершенствования, включая автоматизированный маппинг к формату STIX с помощью LLM моделей, классификацию на основе LLM, нечеткое сопоставление и параллельное кэширование для улучшения покрытия и уменьшения задержек.

Ключевые слова: анализ киберугроз, классификация угроз, интеграция данных из нескольких источников, анализ вредоносных программ, платформа.

Introduction

Cyber Threat Intelligence (CTI) is a structured form of cybersecurity information aimed at providing organizations with critical insights about emerging and existing cyber threats [1]. Such intelligence typically includes data about malicious indicators, tactics, techniques, and procedures used by cyber adversaries, enabling organizations to proactively manage their cybersecurity posture in alignment with internationally recognized frameworks such as the National Institute of Standards and Technology (NIST) Cybersecurity Framework [2], and the EU Directive on security of network and information systems [3].

As the scale of cyberattacks continues to escalate, so does the urgency to respond with timely, actionable, and shareable threat intelligence. According to recent statistics, in 2024 alone there were over 6.06 billion malware attacks globally [4], while on average it takes 194 days to identify a data breach [5] and the Cam4 case holds the record for the largest data breach of all time with over 10 billion compromised accounts [6].

Sharing CTI between organizations significantly enhances their collective defense capabilities and situational awareness [7], [8]. When one entity experiences a cyber-attack, the knowledge gained can aid other entities in managing similar cyber-threats [1], [9]. To ensure efficient interoperability [10], CTI commonly employs standards for delivery mechanism, such as TAXII [11] and content format such as STIX by OASIS [12] which is aligned with MITRE ATT&CK attack patterns [13]. Some works propose using TAXII standard with a blockchain has been proposed to ensure privacy, data integrity and interoperability in CTI sharing [14], [15]. However, despite the rise of threat intelligence platforms (TIPs), most organizations fail to operationalize CTI effectively across their infrastructure. The interoperability problem is well-documented in the literature. Rantos et al. [10] provided a comprehensive account of interoperability challenges in CTI ecosystems, identifying syntactic mismatches, semantic inconsistencies, and governance-level fragmentation as persistent barriers. They emphasize that CTI is not only difficult to exchange but also hard to align across heterogeneous systems. This reveals not only a technical gap, but a deeper systemic failure in CTI integration and interoperability across diverse sources and systems.

To address these gaps, we propose the MFCTIIF — Multi-Feed Cyber Threat Intelligence Integration Framework, a modular system designed to unify multi-source CTI and enhance its operational value. The contributions of this work are as follows:

1. Unified multi-source integration for heterogeneous indicators. MFCTIIF automatically ingests and harmonizes indicators from multiple CTI feeds [16], [17], [18] addressing the heterogeneity challenge.

2. Metadata enrichment for effective risk prioritization by aggregating auxiliary attributes such as AV detections, malware family classifications, and contextual threat tags, MFCTIIF enriches sparse raw indicators.

3. Automated threat-level assessment and actionable CTI output. MFCTIIF incorporates a lightweight analytics module that computes a threat severity score based on frequency and classification of observed malware samples.

4. Empirical benchmarking on recent malware samples. We evaluate MFCTIIF on the 100 latest malware samples from MalwareBazaar, demonstrating its effectiveness in unifying multi-source CTI, enriching sparse indicators, and providing actionable threat-level assessments in real-world conditions.

By combining multi-source integration, semantic enrichment, and automated threat assessment, MFCTIIF converts fragmented and underutilized threat indicators into coherent, high-value intelligence. This enables security operations centers (SOCs) to improve decision-making and reduce mean time to response (MTTR).

The remainder of this paper is organized as follows. Section 2 presents the methodology, beginning with the mathematical model of multi-feed CTI integration (Section 2.1) and the system architecture (Section 2.2), followed by detailed descriptions of the import (Section 2.3), data matching (Section 2.4), analytics (Section 2.5), and export modules (Section 2.6). Section 3 provides the evaluation, including measurements of average API response time (Section 3.1), end-to-end latency per sample (Section 3.2), and analysis of sample characteristics (Section 3.3), threat level calculations (Section 3.4), and family distribution (Section 3.5). Section 4 presents the discussion, highlighting the key findings (Section 4.1) and outlining directions for future research (Section 4.2). Finally, Section 5 concludes the paper and summarizes the key contributions, with references provided at the end.

Methodology

This section describes the design and operation of the MFCTIIF framework, covering its mathematical foundation, architecture, and core modules. Section 2.1 introduces the mathematical model, formalizing the mapping of malware samples to hashes, families, and threat levels. Section 2.2 presents the system architecture, outlining the workflow from data ingestion to threat-level output. Sections 2.3–2.6 describe the four core modules: the Import module for multi-feed ingestion, the Data Matching module for metadata alignment, the Analytics module for automated threat assessment, and the Export module for producing structured JSON feeds.

2.1. Mathematical model

To systematically unify heterogeneous cyber threat intelligence (CTI) feeds and support automated threat assessment, we define a formal mathematical model. This model captures malware samples, their associated attributes, probabilistic detection behavior, and temporal evolution, forming the analytical foundation of the MFCTIIF framework. Let the following sets to represent the principal entities in multi-feed CTI analysis:

- M — set of malware samples under analysis;
- S — set of associated signatures (e.g., cryptographic hashes, AV signatures);
- C — set of malware classes (e.g., Virus, Trojan, Worm);
- T = {LOW,MEDIUM,HIGH} — discrete threat severity levels;
- F_m — set of malware families (grouped by behavioral or lineage similarity).

To relate malware samples to their features, we define four key mapping functions:

1. Cryptographic identity: maps each malware sample to its SHA-256 hash, providing a unique identifier for feed unification.

$$h : M \to [0,1]^{256} \tag{1}$$

2. Signature mapping: associates a sample with the set of signatures collected from av engines and external sources, where P is a power set.

$$\sigma : M \to \mathcal{P}(S) \tag{2}$$

3. Family mapping: assigns each sample to one or more malware families, supporting lineage-based context enrichment.

$$\phi : M \to \mathcal{P}(F_m) \tag{3}$$

4. Threat level classification: maps class information to a final severity level, consolidating lineage-based context enrichment.

$$\tau \, : \, \mathcal{P}(C) \rightarrow T \tag{4}$$

Each malware sample $m \in M$ may belong to multiple classes $c_1, c_2, \dots, c_k \in C$. To compute the overall threat level, we take the maximum severity across all its classes:

$$\tau(c) = \max_{x \in c} \theta(x) \tag{5}$$

Where $\theta \, : \, C \rightarrow T$ is a class-to-severity mapping, for example:

$$\theta(x) = \begin{cases} \text{High}, & x \in \{\text{Virus}, \text{Rootkit}, \text{HackTool}, \dots\} \\ \text{Medium}, & x \in \{\text{Hoax}, \text{Dialer}, \text{RiskTool}, \dots\} \\ \text{Low}, & x \in \{\text{Spam}, \text{Server-FTP}, \dots\} \\ \text{Unknown}, & \text{otherwise} \end{cases} \tag{6}$$

Then, given a sample which belongs to Adware, RiskTool, and Rootkit, we have:

$$\tau(\{\text{Adware}, \text{RiskTool}, \text{Rootkit}\}) = \max(\text{Low}, \text{Medium}, \text{High}) = \text{High} \tag{7}$$

To incorporate detection reliability, we define the detection probability (8) for a malware sample m by an AV engine v. This allows for risk quantification based on detection consensus, confidence-weighted threat scoring and integration with automated analytics.

$$p(m, v) = \frac{d_{pos}(m,v)}{d_{total}(m,v)} \tag{8}$$

Where d_{pos} is the number of positive detections and d_{total} is the total number of scans.

For each sample m, a unified threat feed entry is generated encapsulating its main attributes, such as cryptographic hash, signature, family mapping and threat classification:

$$f(m) = \big(h(m), \sigma(m), \phi(m), \tau(\sigma(m)), p(m, v)\big) \tag{9}$$

This tuple serves as the canonical representation of a threat indicator in the MFCTIIF framework. The complete threat feed is then expressed as:

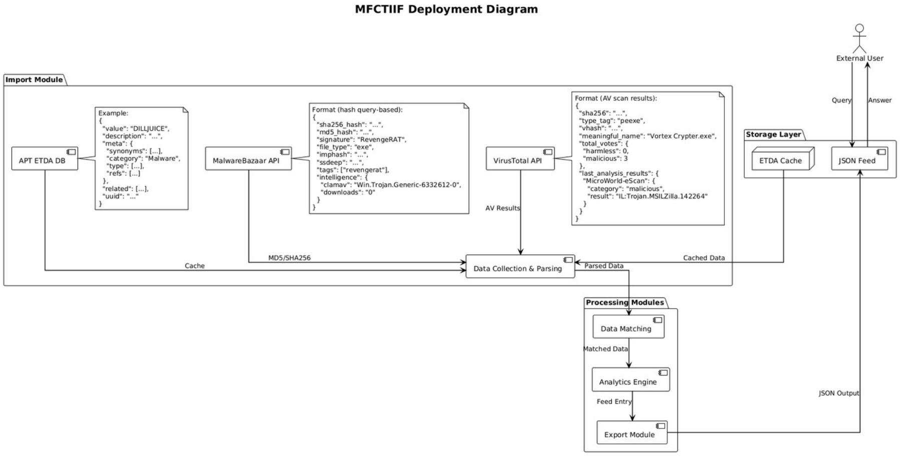
$$F = \{ \ f(m) \mid m \in M \ \} \tag{10}$$

Finally, combining the above, the multi-feed CTI integration model is represented as:

$$\text{MFCTIIF} = (M, S, C, T, F_m, h, \sigma, \phi, \tau, p, F) \tag{11}$$

2.2. System architecture

The system architecture operationalizes the mathematical model introduced in the previous section by implementing a complete multi-stage CTI processing pipeline. Each stage in the workflow corresponds to a function of the model: cryptographic identification, classification and enrichment, threat assessment, and final feed export. Figure 1 shows the deployment diagram of the MFCTIIF, illustrating the interaction between external data sources, internal processing modules, the cache, and the output feed.



```
Example:
{
  "value": "DILLJUICE",
  "description": "...",
  "meta": {
    "synonyms": [...],
    "category": "Malware",
    "type": [...],
    "refs": [...]
  },
  "related": [...],
  "uuid": "..."
}
```

```
Format (hash query-based):
{
  "sha256_hash": "...",
  "md5_hash": "...",
  "signature": "RevengeRAT",
  "file_type": "exe",
  "imphash": "...",
  "ssdeep": "...",
  "tags": ["revengerat"],
  "intelligence": {
    "clamav": "Win.Trojan.Generic-6332612-0",
    "downloads": "0"
  }
}
```

```
Format (AV scan results):
{
  "sha256": "...",
  "type_tag": "peexe",
  "vhash": "...",
  "meaningful_name": "Vortex Crypter.exe",
  "total_votes": {
    "harmless": 0,
    "malicious": 3
  },
  "last_analysis_results": {
    "MicroWorld-eScan": {
      "category": "malicious",
      "result": "IL:Trojan.MSILZilla.142264"
    }
  }
}
```

Figure 1 - MFCTIIF deployment diagram

DOI: https://doi.org/10.60797/IRJ.2026.163.57.1

The architecture is composed of four core modules, such as Import, Data Matching, Analytics, and Export, which are connected in a linear workflow with feedback through caching and storage components. An external user interacts only with the resulting JSON feed, while the internal modules orchestrate collection, enrichment, analysis, and output in sequence.

2.3. Import module

The workflow begins with the Import Module (Figure 2), which collects malware samples from multiple sources, including MalwareBazaar, VirusTotal, and the APT ETDA database. For each batch of new samples, the system first checks the ETDA cache to avoid redundant queries. If the requested data is unavailable, fresh metadata is retrieved from the ETDA database and stored in the cache for subsequent lookups. Simultaneously, the module queries VirusTotal to gather AV scan results for each sample. This stage ensures that every sample entering the pipeline has both its cryptographic identity and associated threat intelligence collected before handing the samples to the next stage.

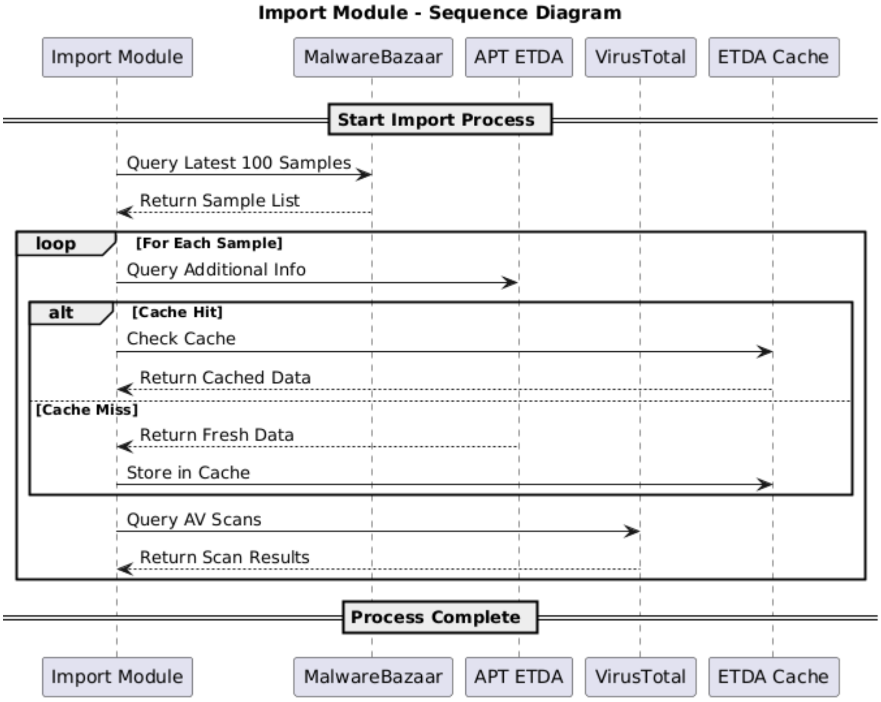


Figure 2 - Import module sequence diagram

DOI: https://doi.org/10.60797/IRJ.2026.163.57.2

2.4. Data matching module

Once raw samples are imported, the Data Matching Module (Figure 3) enriches the data context by identifying malware classifications and their synonyms. This stage performs two ETDA queries for each sample: one to classify the malware type and another to retrieve synonym relationships. Both queries follow the same cache-first strategy — cached results are used when available, and new results are fetched and stored when cache misses occur. After the ETDA lookups, the module performs internal processing to map malware classes, create family lists, and associate AV detections with the appropriate families — forming the core feed entry for the next stage.

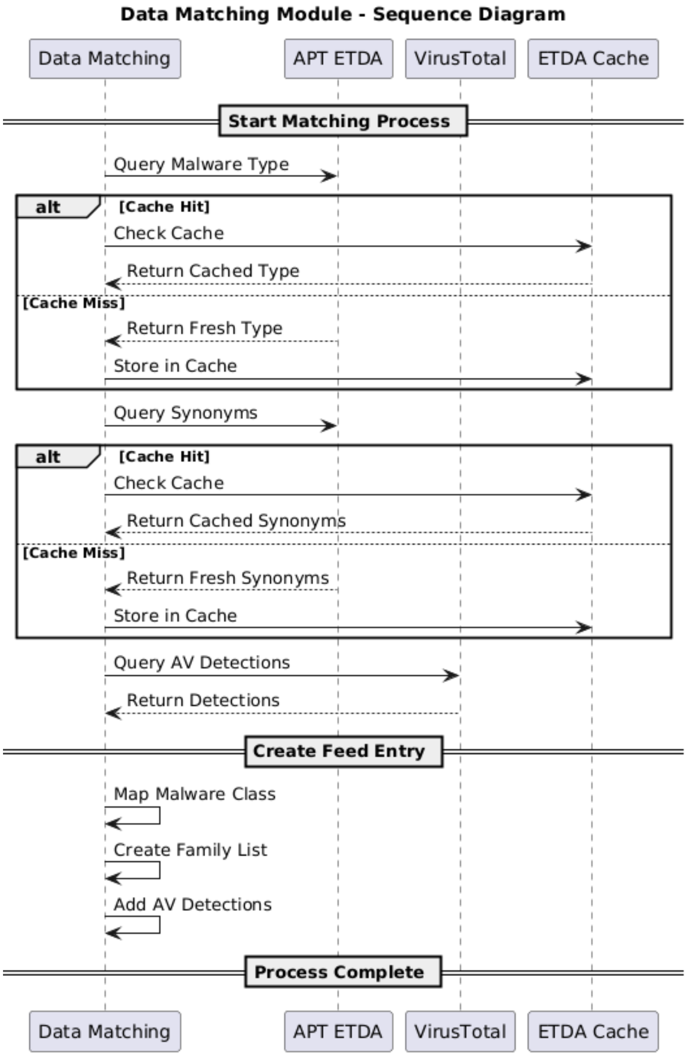


Figure 3 - Data matching module sequence diagram
DOI: https://doi.org/10.60797/IRJ.2026.163.57.3

2.5. Analytics module

The Analytics Module (Figure 4) performs threat level assessment based on the enriched feed entries provided by the Data Matching Module. The core logic focuses on computing the highest observed threat level among the collected indicators. Counters are initialized for high, medium, and low threat categories, each incremented based on the classifications in the feed entry. The final threat level corresponds to the maximum observed category and is then attached to the sample record. At the end of this stage, the feed entry is fully enriched with a computed threat level and is ready for export stage.

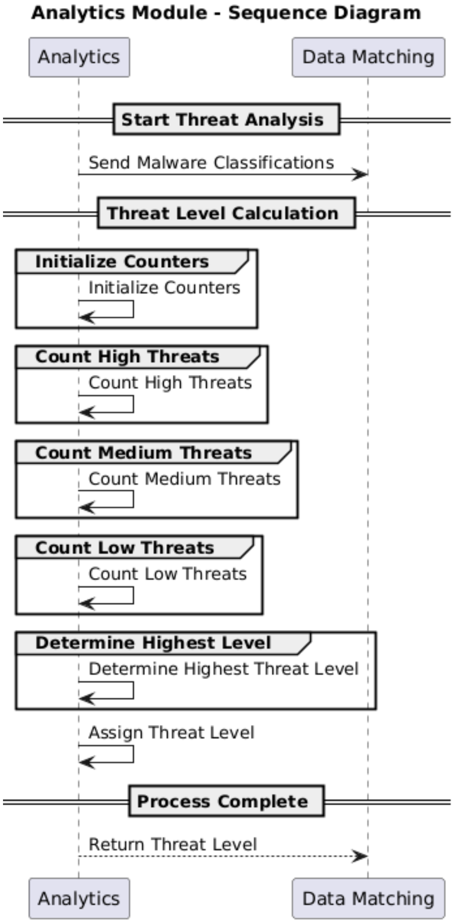


Figure 4 - Analytics module sequence diagram
DOI: https://doi.org/10.60797/IRJ.2026.163.57.4

2.6. Export module

The final stage of the workflow (Figure 5) is handled by the Export Module, which converts enriched feed entries into the JSON format for distribution. Each feed entry is transformed into a JSON object, appended with a newline character, and written to the output file in append mode to preserve history. The module also handles proper file closure and confirmation of successful writes, ensuring that the feed remains consistent and incrementally updateable.

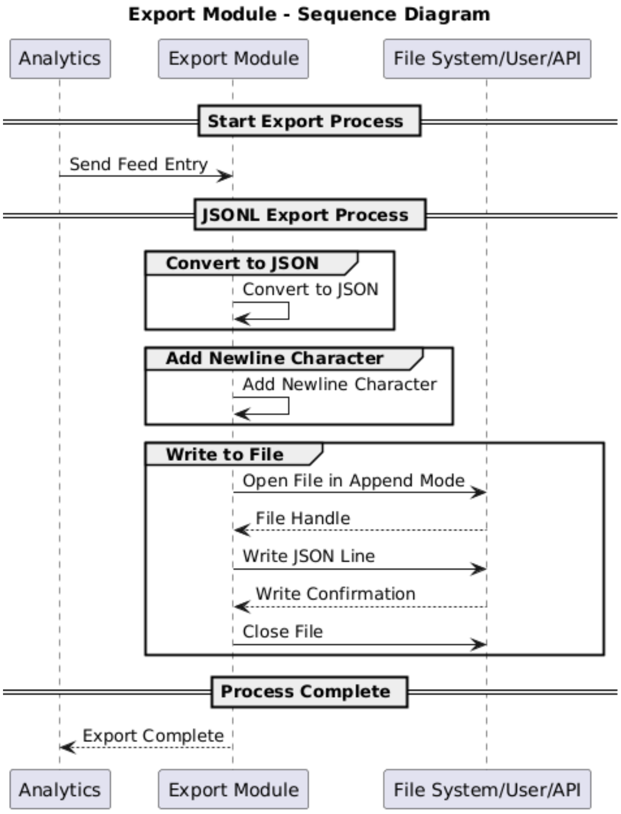


Figure 5 - Export module sequence diagram
DOI: https://doi.org/10.60797/IRJ.2026.163.57.5

Evaluation

This section evaluates the MFCTIIF framework in terms of performance and threat intelligence output quality. Section 3.1 measures the average API response time, reflecting external dependency performance. Section 3.2 reports the end-to-end latency per sample, highlighting workflow efficiency and occasional delays. Section 3.3 analyzes sample characteristics, focusing on metadata coverage in malware_class and malware_family. Section 3.4 presents the threat level distribution, while Section 3.5 examines the malware family distribution.

3.1. Average API response time

Figure 6 illustrates the response times for 10 consecutive queries submitted to the MalwareBazaar’s “get_recent” API. The horizontal axis represents the sequential query index, while the vertical axis shows the measured latency in seconds. Each blue marker denotes the individual response time of a single query, and the connecting line highlights the temporal variation across the series of requests.

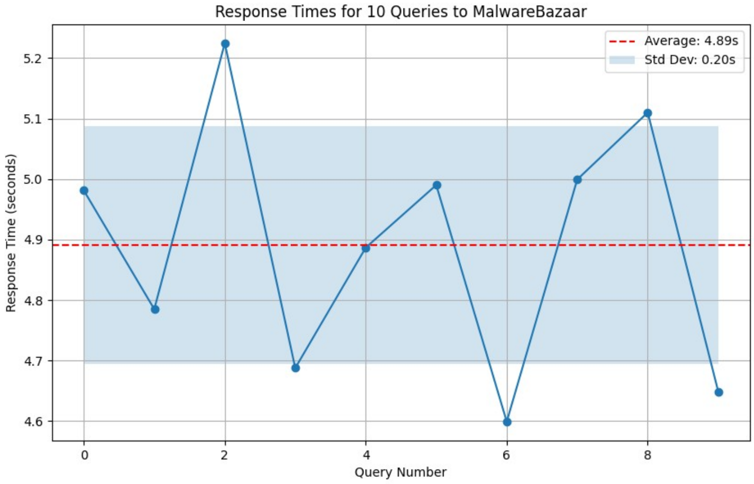


Figure 6 - MalwareBazaar get_recent timings for 100 samples
DOI: https://doi.org/10.60797/IRJ.2026.163.57.6

A red dashed line indicates the mean response time, calculated as 4.89 seconds, providing a baseline for performance assessment. The shaded blue region represents one standard deviation (±0.20 seconds) around the mean, illustrating the natural variation in query latency. However, two notable deviations are observed: Query 2 exhibits the maximum latency (~5.23 s), while Query 6 achieves the minimum latency (~4.60 s). Despite these outliers, the overall trend indicates low variance and a consistent service response, which is beneficial for time-sensitive malware feed processing within the proposed framework.

3.2. End to end latency per sample

Figure 7 shows the end-to-end latency for processing 100 malware samples through the MFCTIIF pipeline. Most samples complete within 15–17 seconds, forming a stable baseline for import, data matching, analytics, and export. This indicates that the workflow is generally efficient and predictable under normal conditions.

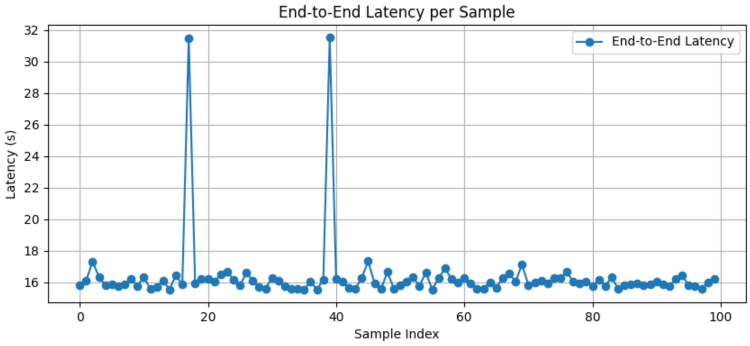


Figure 7 - End to end latency per sample
DOI: https://doi.org/10.60797/IRJ.2026.163.57.7

Two samples exhibit significant latency spikes (~31–32 seconds). These outliers are likely caused by cache misses and VirusTotal API rate-limiting, which introduce long delays when uncached samples require full remote analysis.

3.3. Samples characteristics

Figure 8 illustrates the distribution of the latest 100 malware samples based on the presence or absence of metadata entries in two key fields: malware_class and malware_family. These fields are critical for downstream analytics, particularly for the threat level assessment component of the system.

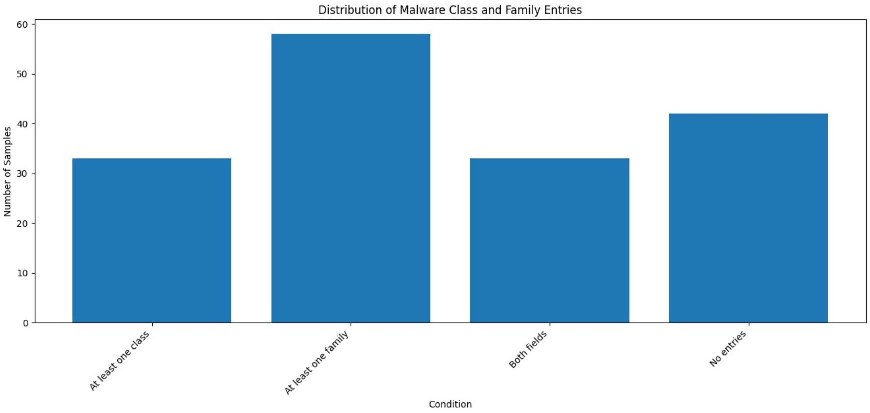


Figure 8 - Sample characteristics distribution
DOI: https://doi.org/10.60797/IRJ.2026.163.57.8

The results reveal that 58% of the samples contain at least one family entry, while 33% contain at least one classification (class) entry. Notably, only one-third (33%) of the samples contain both types of metadata. Conversely, a significant portion of 42% of samples contains neither classification nor family information.

3.4. Threat level calculation

The distribution shown in Figure 9 is a direct consequence of the metadata coverage illustrated in Figure 8. As demonstrated previously, 42% of the analyzed samples lack entries in both the malware_class and malware_family fields, and only 33% contain valid class information required for threat level computation. This absence of classification metadata forces the analytics module to either omit threat scoring or rely solely on fallback detection-to-class mapping.

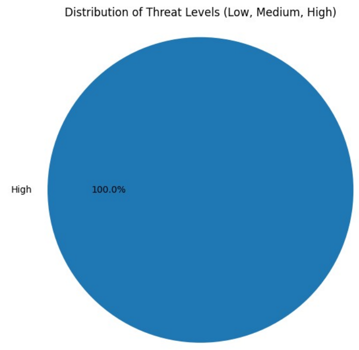


Figure 9 - Threat level distribution in recognized samples
DOI: https://doi.org/10.60797/IRJ.2026.163.57.9

As a result, the system successfully assigns a “High” threat level to all samples with recognized classes, because these samples correspond to well-known high-impact families such as trojans, backdoors, and ransomware. However, the lack of medium and low classifications is not necessarily an indication that the dataset lacks less severe threat — it instead reflects the structural limitation imposed by incomplete metadata.

3.5. Family Distribution

Finally, Figure 10 illustrates the distribution of malware families within the analyzed dataset, highlighting both prevalent and less common threats. The most dominant malware family in the dataset is QuasarRAT (30%), which is a well-known remote access trojan (RAT) capable of persistent system compromise and exfiltration of sensitive data. Its high prevalence signals that remote access threats remain a critical risk vector in the analyzed samples. Next up we have njrat (22%), also known as NetWire, is another widespread RAT family observed in the dataset. It is modular in design and capable of multiple malicious actions, including credential theft and system takeover, reinforcing the dominance of RATs in the current sample set. Mirai (2%) appears less frequently but is significant due to its history of powering large-scale IoT-based DDoS botnets. Even low prevalence in this dataset is operationally important, as Mirai infections can escalate into network-wide attacks. RemcosRAT and Xorbot (2% each) form a smaller but still notable portion of the dataset. These threats indicate a long tail of active RAT and botnet variants circulating in the wild. Amadey (1%) is among the least represented families in the dataset. Despite its low frequency, its appearance highlights the diverse composition of threats, including loader-type malware.

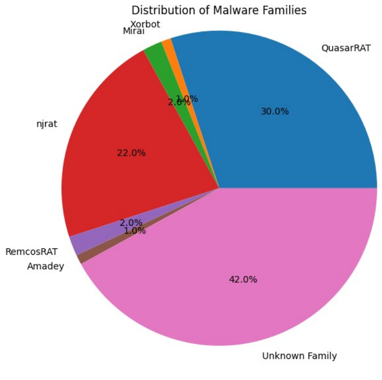


Figure 10 - Malware family distribution
DOI: https://doi.org/10.60797/IRJ.2026.163.57.10

Notably, 42% of samples remain classified as “Unknown Family,” reflecting gaps in available CTI metadata and limitations in system’s current enrichment capabilities.

Discussion

4.1. Key findings

The previous section evaluation of the MFCTIIF highlights its effectiveness in integrating multi-source cyber threat intelligence, while also revealing structural limitations that affect coverage and balance. The system successfully unifies feeds from MalwareBazaar, VirusTotal, and APT ETDA, producing enriched JSON outputs that enable threat assessment and prioritization, as shown in Figure 11.

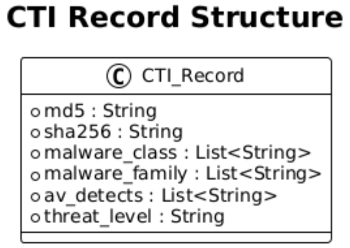

CTI Record Structure
C CTI_Record
o md5 : String
o sha256 : String
o malware_class : List<String>
o malware_family : List<String>
o av_detects : List<String>
o threat_level : String

Figure 11 - Malware sample entry
DOI: https://doi.org/10.60797/IRJ.2026.163.57.11

Across the evaluated dataset of 100 malware samples, the framework consistently classified recognized entries as high-threat malware, dominated by Remote Access Trojans (RATs) and banking trojans. This trend is evident in the malware family distribution, where QuasarRAT (30%) and njrat (22%) emerged as the most prevalent families. Such findings confirm that the analyzed dataset primarily consists of active, high-impact threats, with a strong emphasis on endpoint compromise and data exfiltration.

Metadata coverage, however, proved to be a decisive factor in the system’s overall performance. As shown in Figure 8, 42% of samples lacked entries for both malware class and family, which limited the analytics module’s ability to compute threat levels for the majority of samples. This deficiency directly influenced the threat level distribution presented in Figure 9, where all classified samples were assigned a high-threat level and no medium or low-level classifications were produced. The absence of a graded severity spectrum does not necessarily indicate that the dataset lacks lower-risk samples; rather, it reflects the system’s dependency on complete and consistent metadata. Samples with missing class information, or those requiring fuzzy matching and external enrichment, currently either remain unclassified or rely solely on fallback mapping derived from AV detection strings.

Operational performance analysis further demonstrated that the framework is generally efficient and predictable, with an end-to-end latency of approximately 15–17 seconds per sample under normal conditions. Two prominent latency spikes (~31–32 seconds) were observed and are attributable to VirusTotal cache misses and API rate-limiting, which are known constraints when processing uncached or newly submitted files. Although the baseline performance is acceptable for batch processing, reliance on external APIs introduces non-deterministic delays that could affect real-time alerting and large-scale feed generation.

4.2. Comparative analysis

We evaluate our work and 4 previous related works on 7 attributes, such as

1) multi-Source CTI Integration;

2) heterogeneous Feed Support (MalwareBazaar / VT / ETDA);

3) metadata Enrichment (Families + Signatures);

4) automated Threat-Level Classification;

5) standardized / Actionable Output (JSON/STIX/KG);

6) latency / Performance Evaluation;

7) empirical Evaluation on Recent Malware Samples, as shown in Table 1.

Rastogi et al. [19] in MALOnt focuses on semantic enrichment and ontology-driven knowledge graph construction for malware threat intelligence

1) it aggregates data from multiple threat reports;

2) supports general heterogeneous sources, but not specific to MalwareBazaar/VT/ETDA;

3) it also includes malware families, characteristics, attacker groups;

4) but does not assign low/medium/high threat levels;

5) it provides knowledge-graphs only, not JSON or STIX output;

6) neither does it perform timing nor latency analysis;

7) finally, it is evaluated on annotated reports, not live malware feeds;

Okazaki et al., 2024 [20] system integrates multiple AV engines for collaborative detection using VirusTotal:

1) it supports collaborative multi-AV integration;

2) and has partial support for VirusTotal and MalwareBazaar;

3) metadata focus is on AV voting, not family or signature enrichment;

4) it outputs malicious/benign verdict, but has no severity scoring;

5) it provides detection result only, not JSON/STIX output;

6) it collects recall and weighted voting performance, but not latency.

7) it drives recall evaluation on real 7-day continuous sets of samples from MalwareBazaar.

Gao et al., 2024 [21] ThreatKG uses AI techniques to unify structured/unstructured OSCTI and construct a threat knowledge graph (KG) with rich context:

1) it aggregates OSCTI from many sources;

2) but has no VT/MB/ETDA ingestion;

3) it extracts TTPs, entities, relations;

4) and has no explicit threat level scoring;

5) it uses knowledge graph and no JSON/STIX;

6) finally no runtime nor latency evaluation and;
7) no per-sample malware feed evaluation.

Rastogi et al., 2023 [22] TINKER captures multi-source OSCTI and also builds a CTI Knowledge Graph (CTI-KG) like ThreatKG:

1) it aggregates multi-modal OSCTI (blogs, reports, CVEs, GitHub feeds);
2) but has no direct API-level support for MB/VT/ETDA;
3) enrichment with entities, malware families, relationships;
4) but no threat level scoring;
5) it uses knowledge graph only, no JSON or STIX export;
6) nor does it run latency or runtime evaluation;
7) evaluated on curated CTI reports and triple inference, not live malware feeds.

Table 1 - Comparative analysis table

DOI: https://doi.org/10.60797/IRJ.2026.163.57.12

Ref.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
[19]	+	-	+	-	KG	-	-
[20]	+	+	-	-	-	+	+
[21]	+	-	+	-	KG	-	-
[22]	+	-	+	-	KG	-	-
MFCTIIF	+	+	+	+	JSON	+	+

Despite demonstrating functional success in Table 1 over related works, MFCTIIF exhibits several current limitations. The framework is constrained by incomplete metadata, which produces gaps in class and family resolution, and leads to a skewed threat level distribution dominated by high-severity classifications. External API dependencies introduce sporadic latency spikes, reducing throughput and predictability. Also, the reliance on signature-driven classification and exact CTI mappings limits its ability to handle emerging, obfuscated, or zero-day malware, leaving a portion of samples unclassified and reducing the feed's overall coverage. To address these issues, we propose some future research directions below.

4.3. Directions for future research

We now highlight several directions for future research that can enhance the capabilities, coverage, and operational resilience of the MFCTIIF. These directions focus on improving metadata completeness, classification accuracy, and system performance, thereby addressing the current constraints of metadata gaps, skewed threat distributions, and external API latency.

First, standardizing output using STIX (Structured Threat Information eXpression) will improve interoperability with existing CTI platforms and automated threat-sharing ecosystems. By expressing threat indicators, malware families, and associated attributes in STIX format, the framework could seamlessly integrate with TAXII servers and external SOC tooling. Standardization would also facilitate structured reasoning over indicators and relationships in downstream pipelines [12], [23]. In future iterations, automated mapping to STIX could be supported by LLM-based models, which have demonstrated strong performance in extracting CTI entities and generating machine-readable threat reports, as shown in AZERG [24].

Second, classification accuracy and coverage can be improved by integrating machine learning (ML) and large language models (LLMs) to infer malware classes and threat levels for samples with missing or incomplete metadata. LLMs have recently shown strong utility in CTI contexts, such as classifying threat reports and enriching indicators with contextual labels [25], [26]. In combination with heuristic and keyword-based extraction from AV detections, these methods can address the classification imbalance observed in Figures 8 and 9, producing a more nuanced distribution of low, medium, and high-threat samples.

Third, system performance and scalability can be enhanced through improved caching strategies and task parallelization. Our latency benchmarks demonstrate that end-to-end processing is dominated by the import stage due to VirusTotal rate-limiting. Leveraging persistent caching and parallel execution of API requests could reduce latency spikes and improve batch throughput, a technique widely applied in high-performance data pipelines.

Finally, the adoption of fuzzy matching and similarity metrics can address structural gaps in metadata mapping. Current ETDA lookups depend on exact or near-exact signature matches, leaving many samples unclassified. Implementing Levenshtein distance [27] or Python's difflib ratio [28] for family and signature mapping can significantly improve coverage by detecting misspellings, minor variations, or obfuscated names in threat feeds [29]. Extending this approach to AV detection strings could further enhance threat scoring for previously "unknown" samples.

Conclusion

This paper presented the MFCTIIF — Multi-Feed Cyber Threat Intelligence Integration Framework, designed to address interoperability and enrichment challenges in multi-source cyber threat intelligence. By integrating feeds from MalwareBazaar, VirusTotal, and APT ETDA, the framework performs automated data matching, family and class resolution, and threat-level assessment, producing structured JSON outputs suitable for SOC operations. Evaluation on 100 recent malware samples demonstrated that the system reliably identifies high-risk threats, dominated by RATs and banking trojans, with a stable 15–17 second end-to-end latency and occasional API-induced spikes.

A comparative analysis against four representative CTI aggregation and enrichment frameworks highlights that MFCTIIF is the only approach covering all seven evaluation attributes, including multi-feed integration, heterogeneous feed support, metadata enrichment, automated threat scoring, actionable outputs, latency evaluation, and empirical testing on live malware samples.

Despite these promising results, the framework's current impact is limited by metadata gaps, classification imbalance, and external API dependencies, which leave a portion of samples unclassified or skew threat scores toward high severity. Future enhancements should focus on STIX/TAXII standardization, ML/LLM-driven classification for unknown samples, improved caching and parallelization, and fuzzy metadata matching to expand coverage and reduce latency. These improvements will enable MFCTIIF to evolve into a low-latency, AI-augmented CTI pipeline, further strengthening its value for operational cybersecurity and threat response.

Конфликт интересов

Не указан.

Рецензия

Сообщество рецензентов Международного научно-исследовательского журнала
DOI: https://doi.org/10.60797/IRJ.2026.163.57.13

Conflict of Interest

None declared.

Review

International Research Journal Reviewers Community
DOI: https://doi.org/10.60797/IRJ.2026.163.57.13

Список литературы на английском языке / References in English

1. Johnson C. Guide to cyber threat information sharing / C. Johnson, L. Badger, D. Waltermire [et al.]. — Gaithersburg: U.S. Department of Commerce, National Institute of Standards and Technology, 2016. — URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-150.pdf (accessed: 18.07.2025).

2. Pascoe C. The NIST Cybersecurity Framework (CSF) 2.0 / C. Pascoe, S. Quinn, K. Scarfone. — Gaithersburg, MD : National Institute of Standards and Technology, 2024. — URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=957258 (accessed: 18.07.2025).

3. Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) (Text with EEA relevance). — 2022. — URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32022L2555 (accessed: 18.07.2025).

4. Number of malware attacks per year 2023. — URL: https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/ (accessed: 29.07.2025).

5. Cost of a data breach 2024 | IBM. — URL: https://www.ibm.com/reports/data-breach#/pdf (accessed: 29.07.2025).

6. Largest data breaches worldwide 2025. — URL: https://www.statista.com/statistics/290525/cyber-crime-biggest-online-data-breaches-worldwide/ (accessed: 29.07.2025).

7. Saeed S. A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience / S. Saeed, S.A. Suayyid, M.S. Al-Ghamdi [et al.] // Sensors. — 2023. — Vol. 23. — № 16. — DOI: 10.3390/s23167273.

8. Nweke L.O. Legal Issues Related to Cyber Threat Information Sharing Among Private Entities for Critical Infrastructure Protection / L.O. Nweke, S. Wolthusen // 2020 12th International Conference on Cyber Conflict (CyCon). — 2020. — Vol. 1300. — P. 63–78. — DOI: 10.23919/CyCon49761.2020.9131721.

9. Tounsi W. A survey on technical threat intelligence in the age of sophisticated cyber attacks / W. Tounsi, H. Rais // Computers & Security. — 2018. — Vol. 72. — P. 212–233. — DOI: 10.1016/j.cose.2017.09.001.

10. Rantos K. Interoperability Challenges in the Cybersecurity Information Sharing Ecosystem / K. Rantos, A. Spyros, A. Papanikolaou [et al.] // Computers. — 2020. — Vol. 9. — № 1. — P. 18. — DOI: 10.3390/computers9010018.

11. TAXII Version 2.1. — URL: https://www.oasis-open.org/standard/taxii-version-2-1/ (accessed: 19.07.2025).

12. STIX Version 2.1. — URL: https://www.oasis-open.org/standard/stix-version-2-1/ (accessed: 19.07.2025).

13. MITRE ATT&CK®. — URL: https://attack.mitre.org/ (accessed: 23.07.2025).

14. Pahlevan M. Secure and Efficient Exchange of Threat Information Using Blockchain Technology / M. Pahlevan, V. Ionita // Information. — 2022. — Vol. 13. — № 10. — DOI: 10.3390/info13100463.

15. Pahlevan M. Secure exchange of cyber threat intelligence using TAXII and distributed ledger technologies - application for electrical power and energy system / M. Pahlevan, A. Voulkidis, T.-H. Velivassaki // Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES '21). — New York: Association for Computing Machinery, 2021. — DOI: 10.1145/3465481.3470476

16. MalwareBazaar | Malware sample exchange. — URL: https://bazaar.abuse.ch/ (accessed: 18.07.2025).

17. Threat Group Cards: A Threat Actor Encyclopedia. — URL: https://apt.etda.or.th/cgi-bin/aptgroups.cgi (accessed: 18.07.2025).

18. Wazuh. VirusTotal integration — Malware detection · Wazuh documentation. — URL: https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/virus-total-integration.html (accessed: 18.07.2025).

19. Rastogi N. MALOnt: An Ontology for Malware Threat Intelligence / N. Rastogi, S. Dutta, M.J. Zaki [et al.]. — arXiv, 2020. — URL: http://arxiv.org/abs/2006.11446 (accessed: 31.07.2025).

20. Okazaki N. Optimal Weighted Voting-Based Collaborated Malware Detection for Zero-Day Malware: A Case Study on VirusTotal and MalwareBazaar / N. Okazaki, S. Usuzaki, T. Waki [et al.] // Future Internet. — 2024. — Vol. 16. — № 8. — DOI: 10.3390/fi16080259.

21. Gao P. ThreatKG: An AI-Powered System for Automated Open-Source Cyber Threat Intelligence Gathering and Management / P. Gao, X. Liu, E. Choi [et al.]. — arXiv, 2024. — URL: http://arxiv.org/abs/2212.10388 (accessed: 31.07.2025).

22. Rastogi N. TINKER: A framework for Open source Cyberthreat Intelligence / N. Rastogi, S. Dutta, M.J. Zaki [et al.]. — arXiv, 2023. — URL: http://arxiv.org/abs/2102.05571 (accessed: 31.07.2025).

23. Mahardhika Y. Implementation of Cyber Threat Intelligence on Intrusion Detection System using STIX Framework / Y. Mahardhika, F. Astika, I. Syarif [et al.] // Journal of Artificial Intelligence and Software Engineering (J-AISE). — 2025. — Vol. 5. — P. 205. — DOI: 10.30811/jaise.v5i1.6518.

24. Lekssays A. From Text to Actionable Intelligence: Automating STIX Entity and Relationship Extraction / A. Lekssays, H.T. Sencar, T. Yu. — arXiv, 2025. — URL: http://arxiv.org/abs/2507.16576 (accessed: 29.07.2025).

25. Clairoux-Trepanier V. The Use of Large Language Models (LLM) for Cyber Threat Intelligence (CTI) in Cybercrime Forums / V. Clairoux-Trepanier, I.-M. Beauchamp, E. Ruellan [et al.]. — arXiv, 2024. — URL: http://arxiv.org/abs/2408.03354 (accessed: 29.07.2025).

26. Alam M.T. CTIBench: A Benchmark for Evaluating LLMs in Cyber Threat Intelligence / M.T. Alam, D. Bhusal, L. Nguyen [et al.]. — arXiv, 2024. — URL: http://arxiv.org/abs/2406.07599 (accessed: 30.07.2025).

27. Levenshtein distance // Wikipedia. — 2025. — URL: https://en.wikipedia.org/w/index.php?title=Levenshtein_distance&oldid=1301975018 (accessed: 29.07.2025).

28. Zhang K. A Tutorial for Difflib — A Powerful Python Standard Library to Compare Textual Sequences / K. Zhang. — 2024. — URL: https://medium.com/@zhangkd5/a-tutorial-for-difflib-a-powerful-python-standard-library-to-compare-textual-sequences-096d52b4c843 (accessed: 29.07.2025).

29. Zhang S. Research on string similarity algorithm based on Levenshtein Distance / S. Zhang, Y. Hu, G. Bian // 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). — 2017. — P. 2247–2251. — DOI: 10.1109/IAEAC.2017.8054419.