КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.33.1156
UDC 004.056.5:004.89

Danylo Andreiev
Master's Student,
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0009-0009-7908-5388
dandre-ipt24@lll.kpi.ua

Anatolii Chornyi
PhD Student,
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0009-0001-4147-9084
anacho-ipt23@lll.kpi.ua

Iryna Stopochkina
PhD in Engineering Sciences, Associate Professor,
Associate Professor at the Department of Information Security
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0002-0346-0390
i.stopochkina@kpi.ua

Mykola Ilin
PhD in Engineering Sciences, Associate Professor,
Associate Professor at the Department of Information Security
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: 0000-0002-1065-6500
m.ilin@kpi.ua

METHODOLOGY FOR AUTOMATING CYBER INCIDENT REPORTS USING LLM

Abstract. The work is devoted to the issues of automation of reporting as part of Threat Intelligence processes. The purpose of the work is to develop a methodology that allows to reduce the burden on employees who process and document the results of cyber incidents in accordance with the requirements of regulatory documents. Among the main results of the work, a reusable instruction template for a large language model (LLM) is proposed. The presented template allows to provide clear instructions, namely required and optional fields, permissible values that are entered into the report fields. Software models based on the Pydantic library are proposed for generating and checking the response in JSON format from LLM. This allows to reduce the length of instructions for LLM by approximately 3 times. A RAG pipeline architecture is proposed to take into account the specific context of regulatory documents in the field of cyber incident reporting. Such a pipeline allows to follow the requirements of legislation and standards without the need to manually prescribe these requirements in the instructions, which speeds up the generation process and improves the quality of reports. A software model has been developed that allows automated generation of a cyber incident report. Such a model does not require manual filling in of incident characteristics, user interaction with the Threat Intelligence platform using the example of MISP (Malware Information Sharing Platform). This approach allows reducing the time for creating a report from hours to minutes, and improving the efficiency of exchanging threat data, avoiding time and financial investments. Another result of the work is a comparative analysis of report generation when using different LLMs, in particular Claude Sonnet 4.5, Gemini 2.5 pro, Grok xAI, GPT 5, DeepSeek, Llama in terms of quality and cost of report generation. For comparison, report quality criteria were proposed, and compliance with the criteria was assessed by an expert method. As a result, the Claude Sonnet 4.5, Gemini 2.5 pro models were identified as leaders in terms of the quality of generated reports. It was established that LLMs are a promising tool for implementation in processing and communication processes in the field of cybersecurity incidents, their use allows fully automating the Threat Intelligence reporting process in an organization.

Keywords: threat intelligence; large language models; cybersecurity incidents; reporting.

INTRODUCTION

With the development of cyberattacks, a proactive response approach is becoming increasingly relevant, which requires correctly prioritizing cyber threats and building defenses based on existing experience. The collection of such data on real threats and incidents for specific companies and organizations falls within the concept of Threat Intelligence. The process of collecting and presenting information itself must comply with the requirements and templates established by legislation. In particular, the following aspects constitute important information: the potential adversary, the adversary's objective, the technical means used, tactical and technical characteristics, and indicators of compromise. Based on such information, infrastructure protection specialists can develop strategic, tactical, and operational decisions that determine the success of building effective defense.

Problem statement. According to current regulatory requirements, the issue of cyber incident reporting for most institutions is shifting from optional to mandatory. However, not all institutions are adequately prepared for such a transition. In addition, high-quality cyber incident reporting requires specially trained professionals who are knowledgeable in the regulatory framework, as well as in cybersecurity methods and tools. With the emergence of a new tool, large language models (LLMs), it has become possible to automate a number of routine human tasks. This study aims to adapt the capabilities of LLMs to the task of timely reporting in an automated mode. A number of works have already taken steps toward search-related tasks, particularly in finding user-relevant textual content and answering questions using LLMs [1], as well as tasks requiring a certain level of precision (for example, binary sample analysis [2]); there are existing technologies that enable solving tasks of this class [3]. Therefore, it is logical to conduct a study devoted to the automation of cyber incident reporting, taking into account regulatory requirements for Threat Intelligence in Ukraine and European countries [4-6].

Analysis of recent research and publications. Study [7] analyzes the cyber incident process and emphasizes the role of Threat Intelligence platforms. The work highlights the complexity and multidimensional nature of implementing proper reporting in the field of cyber incidents. The stages of the process are examined, and challenges of the Cyber Threat Intelligence process are identified, including significant financial and time costs, as well as the need for highly qualified professionals. However, this study does not provide proposals for addressing these challenges.

Research [8] proposes a framework for automated response to various types of cybersecurity incidents. The framework utilizes the capabilities of LLMs, including GPT-4, GPT-4o, and Claude 3.5 Sonnet. However, the work focuses on the general cybersecurity incident response chain without emphasizing Threat Intelligence issues and reporting communication with governmental response teams and information dissemination processes.

Paper [9] demonstrates specific aspects of working with LLMs, where providing precise execution instructions is crucial. The study proposes extending the data context by supplying clarifying information to the LLM to facilitate analysis. Such an approach is advisable to adapt for the needs of cyber incident report generation.

In [10], LLM capabilities are applied to the task of extracting structured data from large-scale multilingual railway accident reports. To solve this task, a RAG pipeline was proposed and integrated with a graph database. The analysis was conducted using GPT-4o mini and Gemini 2.5 Pro models. The approach is promising for report analysis; however, it is not directly related to the domain of cyber incident response, which has its own specific requirements.

With regard to protection against cyber threats, timely information exchange is essential. One such platform is MISP [11]. Alternative solutions also exist; however, according to a 2025 analysis, MISP is widely used. The MISP exchange format is described in [12] and allows for a sufficiently comprehensive description of threats. Alternatives include CYBEX [13] and STIX [14], which have their own specific features. Nevertheless, MISP is a progressive solution that is likely to obtain RFC status in the future. Interaction with such platforms in accordance with established standards requires specially trained professionals, which may pose a problem for enterprises that lack such personnel or have them in limited numbers.

In Ukrainian practice of cyber incident response, the regulatory framework requires compliance with a specific set of requirements [15].

For main cybersecurity actors – including government institutions, sectoral critical infrastructure protection authorities, and critical infrastructure entities – it is necessary to quickly master proper cybersecurity incident reporting. These information exchange rules are based on the TLP 2.0 protocol [16]. A defined sequence of actions is established for detecting a cyber incident and subsequently informing relevant parties, including reporting procedures. The form of the cyber incident / cyberattack notification card is specified, and its completion requires certain experience and time expenditure. For many enterprises classified as critical infrastructure entities, there may be a lack of time and qualified specialists to prepare such documentation. However, automated systems or tools for generating such reports are currently not provided.

The task of automating cyber incident reporting and platform-based reporting requires specifying precise values and adhering to a defined data entry format for incident information. LLM providers have proposed valid solutions, which are largely similar to each other. The concept of "structured output" is described in source [17].

This study is aimed at overcoming the aforementioned challenges and shortcomings.

Research objective. The objective of this study is to develop an automated pipeline for generating the necessary documentation and reporting information within the Threat Intelligence approach, based on LLM capabilities and regulatory requirements. The stated objective is achieved by solving the following tasks:

- To build a reporting model based on the recommendations of the widely recognized document NIST 800-61 Rev.2 and Ukrainian legislation for further automation;
- To propose a method for generating an incident report based on a report-generation pipeline architecture using a vector database and context extension technology based on the MCP protocol;
- To propose an architectural model of a cyber incident report generation system;
- To develop criteria for the comparative analysis of different LLM models for incident report generation.

RESEARCH METHODOLOGY

The developed AI agent software solutions consist of several components, each performing its own function. To achieve maximum program efficiency, it is necessary to properly select the technical solutions to be used, taking into account compatibility with the chosen programming language. The AI agent was implemented in Python 3 using a set of libraries. The following components were also applied:

1. A vector database.
2. An embedding model.
3. A large language model serving as the core of the application.
4. A set of required Python libraries.

Since the application implements a Retrieval-Augmented Generation (RAG) pipeline to obtain data from regulatory legal acts of Ukraine and the European Union, a solution was selected for storing the embeddings of these documents, which are generated by a dedicated model.

The RAG pipeline consists of the following stages [18]:

1. Retrieval, during which the model accesses an external information source and retrieves the necessary data.
2. Augmentation, which involves adding context to the retrieved information, for example, metadata.
3. Generation, which uses the obtained data as additional context when generating the final results.

This approach is also applied in the system to ensure better compliance of the generated results with the regulatory requirements of Ukraine and the European Union.

During the operation of the RAG pipeline, the LLM operates on embeddings. These represent fragments of text as vectors of a fixed length. The transformation of text into embeddings is performed by a separate embedding model. This makes it possible to conduct semantic search and return results that are most semantically relevant to the query.

Accordingly, special requirements are imposed on the database that stores and processes embeddings, which necessitates the use of specialized solutions. The main requirements include:

- Optimized Approximate Nearest Neighbors (ANN) search.
- Metadata filtering.
- Hybrid search (a combination of multiple search types and filters).
- Re-indexing (recreating the index with different parameters).

Despite the fact that modern relational and non-relational databases support vector storage, they are not suitable for organizing a full-fledged RAG pipeline and interaction with large language models.

For the application, ChromaDB was selected. Although it performs poorly with large volumes of data, has limited scalability, and lacks many features available in other solutions, it allows rapid deployment and is well integrated with Python and the LangChain library. This database is well suited for small datasets, rapid prototyping, and proof-of-concept (PoC) development, making it an appropriate choice for implementing a basic AI agent.

To generate embeddings, the model multilingual-e5-base was selected. Although this model does not demonstrate the highest benchmark performance on MTEB metrics, it has several advantages that enable its effective use:

1. The model is open-source and can be deployed locally free of charge. This ensures data privacy and independence from third parties.

2. The model size is relatively small, allowing deployment on a local machine without significant memory requirements.

3. Although it underperforms compared to models provided by major vendors, it still delivers solid results, outperforming, for example, paraphrase-multilingual-MiniLM-L12-v2.

RESEARCH RESULTS

Report Model Development.

A report model was developed in accordance with:

a. Ukrainian legislation; and
b. NIST 800-61 Rev.2.

To create the report model based on Ukrainian legislation, a number of regulatory documents governing the procedure for cyber incident reporting were used. These documents were reviewed in Section 1.3 of this study. The primary document is Order of the State Service of Special Communications and Information Protection of Ukraine No. 570 dated 03.07.2023 "On Approval of Methodological Recommendations for Response by Cybersecurity Entities to Various Types of Events in Cyberspace" [15]. In particular, its appendices contain the Cyber Incident / Cyberattack Notification Card.

A mapping was developed between the field names in the incident card, the corresponding fields in the report model, the possible field values, and the explanations for each field.

For example, see Table 1.

Table 1

Mapping of Incident Card Fields to the Report Model

Field Name in the Card	Corresponding Field in the Report Model	Possible Values (if limited)	Field Description
Object of the cyberattack	attack_objects	-	Includes fields such as type, name, operating system, time zone, network settings, user accounts, CVE (if any), and conclusion
Has the cyber incident / cyberattack been resolved?	incident_solved	Yes, No	Indicates whether the incident handling process has been completed
Is CERT-UA assistance required?	cert_ua_help_needed	Yes, No	Indicates whether assistance from CERT-UA is required to resolve the incident
Has the cyber incident / cyberattack been reported to other primary cybersecurity entities? If so, which ones?	reported_to_other_subjects, other_subjects_list, other_subjects_comment	Yes, No; Security Service of Ukraine; Ministry of Defense of Ukraine and the General Staff of the Armed Forces of Ukraine; Intelligence agencies; etc.	Indicates whether the incident was reported to other entities in Ukraine and specifies which ones

This model can now be used to solve three tasks simultaneously:

1. Generating instructions for the LLM to obtain results in a predefined format;
2. Validating LLM responses and generating errors in cases of invalid field values;
3. Using the resulting JSON for automatic template completion and generation of a finalized PDF report.

The LLM receives instructions, a prompt, and a textual description of the cyber incident. It then analyzes the description and populates the fields according to the defined schema. A separate mechanism is also предусмотрено for fields that include the option "Other." If the LLM determines that the predefined values are not suitable for a given field, it assigns the value "Other" and fills in the corresponding fieldname_other field at its own discretion. For example, if the incident was detected by a SIEM system (as stated in the incident description), the model will specify this information in the detection_source_other field.

The model containing the fields recommended by NIST SP 800-61 Rev.2 was created based on Appendix B of this document (referred to in the document as "data elements"). Although it is stated that the list is not exhaustive and that each organization should develop its own set of fields according to the specifics of its processes, it can serve as a foundation for future extensions of this automated solution.

In this context, it is advisable to use graph-based models [19], in particular those built with LangGraph. Each node performs a specific operation; for example, a node can be created for generating the core of the report, and the graph can then be expanded by adding nodes to address new requirements. In the document, the fields are divided into two groups: Basic Data Elements and Incident Handler Data Elements. The first group contains general information about the cyber incident, while the second group is intended to be completed by the specialist directly handling the incident and includes fields such as the current incident status, cost (damage incurred), root cause of the incident, and others. Two separate models describing these groups were created; however, during program execution, they are combined into a single JSON structure for convenience. An example of the mapping between the incident card and the Pydantic-based report model suitable for EU countries is presented in Table 2.

Table 2

Comparing the incident card and the model for the report

Field name in the document	Corresponding field in the report model	Field description
Contact Information for the Incident Reporter and Handler	Individual model EUContact (name, role, org_unit, affiliation, email phone, location)	Contact details of the applicant and incident handler (including name, position, email, etc.)
Status change date/timestamps (including time zone)	Individual model EUIncidentTimestamps (incident_start, incident_discovered, incident_reported etc.)	Incident timestamps (when it started, when it was detected, when it was resolved, etc.)
Physical location of the incident	physical_location	Where the incident occurred (city, state, etc.)

In order for LLM to be able to automatically create events in MISP based on the received data and IoC, it is also necessary to create a corresponding model. Since the advanced capabilities of MISP will not be used in the work, it was decided to describe two entities: Event and Attribute.

Report Generation Method Development.

The developed method implements an intelligent incident information processing pipeline that transforms fragmented cyberattack data into a structured report and an event entry in a Threat Intelligence Platform. The method is based on the following principles:

1. Principle of Hybrid Intelligence.

The method combines the capabilities of LLMs and applies a neuro-symbolic approach. The LLM performs the role of a cognitive core for interpreting unstructured data, while the Pydantic library and JSON schemas act as a structural framework that guarantees compliance of the output with technical requirements. Unlike existing approaches, this method reduces model hallucinations.

2. Context-Dependent Generation (RAG).

The use of the Retrieval-Augmented Generation (RAG) architecture ensures the legal and regulatory accuracy of the method. According to the approach, the context is dynamically enriched with up-to-date regulatory documents (of Ukraine or EU countries) retrieved from a vector database directly at the time of report generation. This makes a system based on this method adaptive to legislative changes without requiring model retraining.

3. Multi-format Output.

The method is built on the idea that the same incident must be presented in different formats for different stakeholders:

- For the regulator: An official PDF report (in accordance with the standards of the State Service of Special Communications and Information Protection of Ukraine or the requirements of NIS2).
- For the technical community: Structured data in MISP format, enabling the immediate dissemination of Indicators of Compromise (IoCs).

4. Human-Centered Automation.

A human performs the role of reviewer and decision-maker, while the system takes over routine tasks such as context aggregation and formatting, leaving the final decision to the expert. For example, JSON verification, model selection, and error validation may be performed. This is critically important in cybersecurity, where reporting errors may have legal consequences.

5. Modularity.

Through the use of the Model Context Protocol (MCP) and supporting libraries, the method can be easily integrated into existing SOC (Security Operations Center) infrastructure. It acts as a “linking component” between raw incident data and Threat Intelligence Platforms, ensuring seamless data transformation and exchange.

ISSN 2663 – 4023

Architecture Model Development for an Automated Cyber Incident Report Generation System.

Based on the method requirements, we propose an architecture for an automated report generation system that ensures easy replacement of one set of regulatory documents with another and is capable of operating with minimal user intervention.

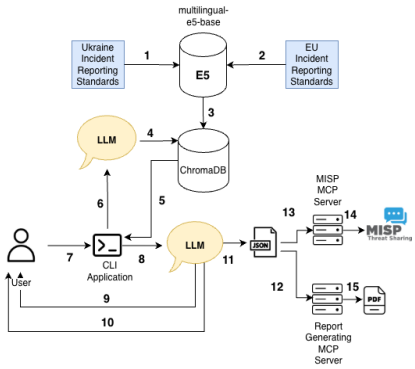

Fig. 1. Model of the architecture of the automated cyber incident report generation system

In the proposed model, the following components can be distinguished:

– Regulatory and legal base for Ukraine and the EU, represented as PDF and HTML files, and an embedding model that converts the texts and stores them in a vector database. This procedure is performed only once.

– Vector database, which stores the embeddings and serves for the RAG pipeline.

– LLM, which generates the final JSON and calls the necessary tools. The LLM is selected at the start of program execution.

– MCP servers for generating PDF reports and MISP events based on the JSON built according to a predefined Pydantic model.

– Operational MISP instance, accessible from the machine on which the application code is executed.

– Auxiliary code, such as parsers for standard texts used for subsequent embedding creation.

The graph of the application that implements the procedural part of the method is shown in Figure 2.

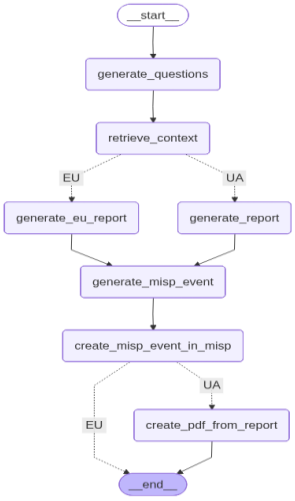

Fig. 2. Program flowchart

The procedural part of the method represents a sequence of actions described below.

The user records data about the cyber incident in a file, which is then read by the program. At the beginning of program execution, the user selects according to which regulatory framework the incident report should be generated (Ukraine or the European Union), as well as the LLM that will be used as the main model. The LLM will receive the data, call tools, and populate the JSON.

The LLM generates a list of queries to the vector database, which will be used in the RAG pipeline to obtain context from the database of documents corresponding to the selected legislation.

The queries generated in the previous step are sent to the vector database, where a search is performed, and the most semantically relevant documents are returned and stored in the graph state.

The user's incident description, along with the retrieved context, is provided to the LLM to generate a JSON that must conform to the predefined schema, which specifies possible fields and their values. After generation, the resulting JSON undergoes validation (without LLM involvement).

If validation fails, the user is notified of the errors found. They decide whether execution can continue with the obtained data or stop to attempt another run or use a different LLM model. In most cases, the errors are non-critical, and the report can be correctly generated based on the resulting JSON.

After this, the execution flow depends on the selected legislation. In the case of Ukrainian legislation, a PDF report is generated. If the EU regulatory framework was chosen at the beginning, this step is skipped, and the resulting JSON is simply presented to the user.

Next, regardless of the selected legislation, a MISP event is generated. Another JSON is created, conforming to a pre-established schema for MISP events and attributes. Using this JSON, an MCP server and the Pymisp library establish a connection to MISP, after which an event is created, and attributes and Indicators of Compromise (IoCs) are added based on the JSON generated by the LLM.

The developed method and architecture model were tested on three incidents with varying levels of detail. JSON files were collected for both the Ukrainian and European branches. An example of a completed JSON for the Ukrainian branch is shown in Figure 3. An example of the attributes extracted by the model from the incident description and added to a MISP event as attributes is shown in Figure 4.

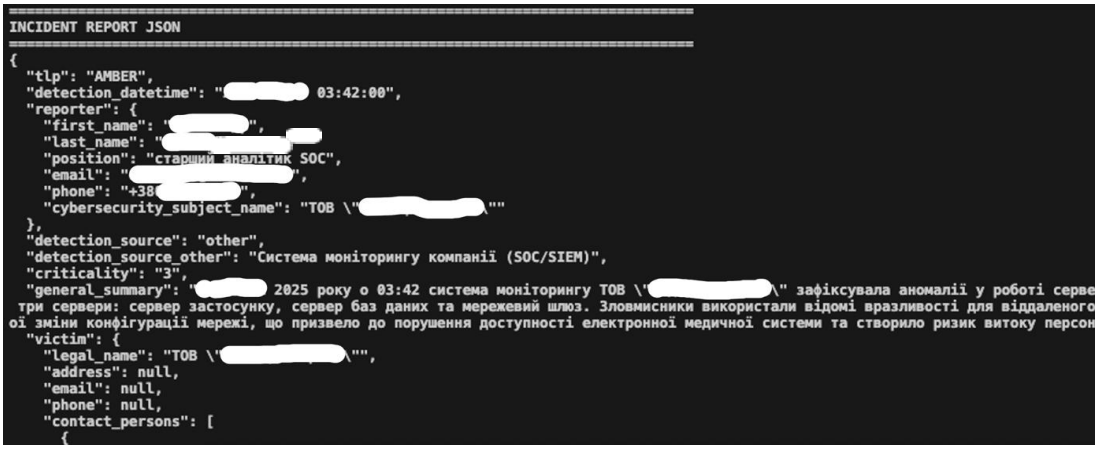

Fig. 3. Example of the JSON received (sensitive information is obscured with white)

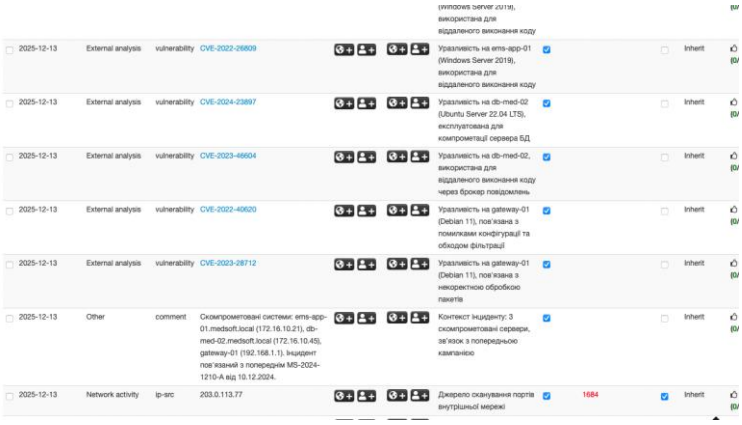

Fig. 4. Example of an event created in MISP

Development of Criteria and Evaluation Results of the Developed Solutions. To assess the quality of the application's performance and compare different language models in the task of generating cyber incident reports based on the user's description and context from the ChromaDB vector database, it is necessary to define clear criteria for evaluating the quality of the obtained results. The evaluation covers the completed PDF report for Ukrainian legislation and the generated JSON for the EU regulatory framework. The created MISP event, along with its attributes and Indicators of Compromise (IoCs), is evaluated for both program operation variants.

The following evaluation criteria are introduced:

1. Report Completion Quality: This criterion evaluates the correctness of the generated data, absence of hallucinations, and the accuracy of representing all information provided by the user in the incident description.
2. Report Completion Completeness: This criterion evaluates how thoroughly the incident is described in the report. The language model should infer and populate incident fields for which explicit data is not provided in the description. However, fields that cannot be filled without the necessary data should remain empty.
3. Correctness of MISP Event Creation: This criterion assesses the accuracy of recognizing Indicators of Compromise in the text, their type (network, host) and category (IP address, hash, etc.). It also evaluates the correctness of comments and notes attached to indicators, MISP event tags, and other data recorded in MISP as a result of the program.
4. Quality of JSON for the EU: This criterion evaluates how correctly the model populated the JSON based on the provided Pydantic schema and instructions according to the requirements in NIST 800-61 Rev.2. Both the correctness of representing the information provided in the incident description and the correctness of inferences for fields such as business impact are considered.
5. Quality of MISP Event for the EU: This criterion evaluates the completeness and accuracy of the MISP event: whether all IoCs were added as attributes, how precise the comments on the indicators are, and whether implicitly stated indicators (e.g., CVE vulnerability identifiers) were added.

Using these parameters allows a comprehensive evaluation of the model's performance and comparison of LLMs in the context of the task.

The models were evaluated according to the above criteria on a scale from 1 to 5. Two types of entities were assessed: PDF report/JSON and MISP event. The scoring for the report (PDF/JSON) is defined as follows:

1. The report is incorrectly filled, or most fields are missing. Critical errors or hallucinations are present; the report does not match the description or requirements.
2. Some fields are filled, but contain significant inaccuracies. Fields not explicitly provided in the description are ignored.
3. Necessary fields are correctly filled based on the incident description. Most fields requiring inference by the model are also correctly populated. Minor inaccuracies may exist but are non-critical.
4. Everything explicitly described in the incident is fully and correctly reflected. For atypical values, the "other" field is used with a correct entry. Inferences are correctly made, and most of these fields are populated.
5. All information from the incident description is correctly reflected. All fields requiring model inference are correctly populated. For atypical values, the "other" field is used with a correct entry. All free-text fields are correctly and thoroughly completed.

The scoring scale for MISP events and attributes is as follows:

1. The model added only a small portion of indicators to the event. The event description does not reflect the essence, and comments on indicators are incorrect or missing.
2. Most indicators were added; descriptions are present but partially incorrect. The event name partially reflects the incident. Additional comments (external analysis or other) are missing.
3. All IoCs explicitly provided in the incident description are correctly added. Comments are present but not always sufficiently informative. The event name reflects the essence of the incident. Additional comments are present but insufficiently informative.
4. All indicators specified in the incident are present with correct comments. The event name fully reflects the incident. Additional comments are present and informative.
5. All explicitly listed indicators are present with informative comments. The event name fully reflects the incident. Indicators inferred by the model, such as CVE vulnerability identifiers, are also added. External analysis or other comments providing additional context are included.

The results and criteria were provided to experts for evaluation. The results were then aggregated for each incident the models worked on, which were conditionally labeled as "light", "medium", and "serious".

The final average ratings from all experts for a "minor" incident are shown in Figure 5.

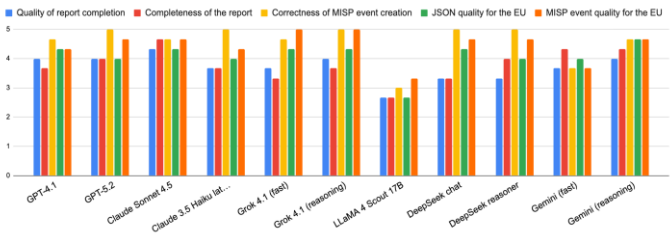

Fig. 5. Comparison of results for a minor incident

Ratings for an “middle” incident are provided in Figure 6.

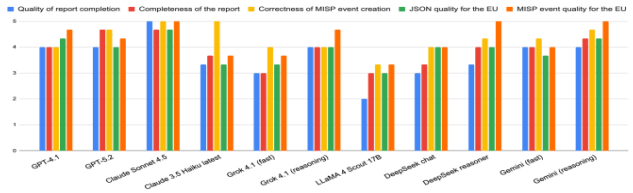

Fig. 6. Comparison of results for a middle incident

The aggregated results for a “major” incident are shown in Figure 7.

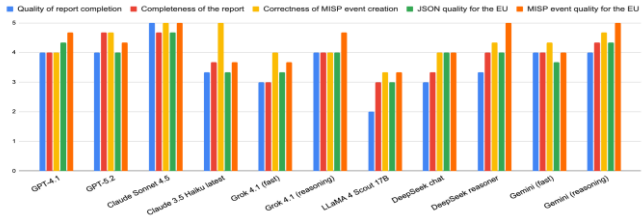

Fig. 7. Comparison of results for a major incident

As additional factors, data were also collected on the number of data validation errors generated by each model, as well as the total resources spent on processing all test incidents. Each model processed an equal amount of data. This information is presented in Table 3.

Table 3

Comparison of errors and costs for models

Model	Number of mistakes during the tests	Total cost for all tests, according to pricing as of December 2025
GPT-4.1	2	$1.37
GPT-5.2	0	
Claude 3.5 Haiku latest	2	$0.21
Claude Sonnet 4.5	1	$0.88
Grok 4.1 (fast)	1	$0.0219
Grok 4.1 (reasoning)	1	$0.0552
LLaMA 4 Scout 17B	3	$0
DeepSeek chat	1	$0.04
DeepSeek reasoner	1	$0.06
Gemini (fast)	0	$1.12
Gemini (reasoning)	0	

In combination with the expert evaluations, this table allows the assessment of the models in the context of generating reports based on the user-provided incident descriptions.

Conclusions and Prospects for Further Research. Within the conducted study, a methodology for automating cyber incident reporting based on the use of large language models (LLMs) and RAG technology was developed and tested. Based on the results, the following conclusions can be drawn:

A unified report model was developed that integrates the requirements of Ukrainian legislation (specifically, DSSZIU Order No. 570) and international standards (NIST 800-61 Rev.2). The created field-matching system allowed structuring data for automated processing.

An architecture of the RAG pipeline using the ChromaDB vector database and the multilingual-e5-base embedding model was proposed. This ensured consideration of the current regulatory context without the need for manually inputted instructions, reducing prompt length by approximately three times.

A software prototype was implemented that automatically generates PDF reports and creates events on the MISP threat intelligence platform. The implementation of Pydantic models ensured strict validation of LLM responses in JSON format.

A comparative analysis of modern LLMs was conducted based on quality, completeness, and generation cost. Expert evaluation identified Claude Sonnet 4.5 and Gemini (in reasoning mode) as leaders in report completion quality and correctness of MISP event creation.

The effectiveness of the approach was demonstrated: automation reduces the time for preparing reporting documentation from several hours to a few minutes, minimizing human factor influence and workload on specialists.

Prospects for further research include expanding the knowledge base, in particular integrating additional international standards (e.g., ISO/IEC 27035) and specific EU sectoral requirements. Another promising direction is the integration with SIEM/SOAR solutions to directly collect logs and alerts from monitoring systems, which would completely eliminate the stage of manual incident description by the user.

REFERENCES

1. Ibrahim, I. M., Soliman, M., & Ossama, S. (2025). Leveraging large language models for document analysis and decision-making in AI chatbots. *Advanced Sciences and Technology Journal*, 2(1), Article 1034. https://doi.org/10.21608/astj.2025.342484.1034
2. Voitsekhovskyi, A., Stopochkina, I., Sun, P., Xie, J., Ilin, M., & Novikov, O. (2026). Detection of vulnerabilities in software for unmanned aerial vehicles by using large language models. *Eastern-European Journal of Enterprise Technologies*, 1(2), 36-47. https://doi.org/10.15587/1729-4061.2026.352029
3. Fezari, M., & Al Dahoud, A. (2026). The evolution of retrieval-augmented generation (RAG) in AI [Preprint]. https://doi.org/10.13140/RG.2.2.27107.62245
4. 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. (2022). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2555
5. Cyber Solidarity Act. (2024). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32025R0038
6. Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). *Computer security incident handling guide (NIST Special Publication 800-61 Rev. 2)*. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-61r2
7. Ainslie, S., Thompson, D., Maynard, S., & Ahmad, A. (2023). Cyber-threat intelligence for security decision-making: A review and research agenda for practice. *Computers & Security*, 132, 103352. https://doi.org/10.1016/j.cose.2023.103352
8. Lin, X., et al. (2025). IRCopilot: Automated incident response with large language models (arXiv:2505.20945) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2505.20945
9. Novikov, O., Ilin, M., Stopochkina, I., Ovcharuk, M., & Voitsekhovskyi, A. (2025). Application of LLM in UAV route planning tasks to prevent data exchange availability violations. *Cybersecurity: Education, Science, Technique*, 1(29), 420–431. https://doi.org/10.28925/2663-4023.2025.29.892
10. Sohi, S., Balan, D., Anjomshoaa, A., & Polleres, A. (2024). Towards harmonised rail safety knowledge: LLM techniques for EU accident report processing. In *CEUR Workshop Proceedings*. https://ceur-ws.org/Vol-4079/short4.pdf
11. MISP Project. (2025). *Features of MISP, the open source threat sharing platform*. Retrieved October 30, 2025, from https://www.misp-project.org/features/
12. Dulaunoy, A., & Iklody, A. (n.d.). *MISP core format*. MISP Standard. https://www.misp-standard.org/rfc/misp-standard-core.html
13. Rutkowski, A., Kadobayashi, Y., & Furey, I. (2010). CYBEX: The cybersecurity information exchange framework. *ACM SIGCOMM Computer Communication Review*, 40(5).
14. OASIS Open. (n.d.). *STIX introductory walkthrough*. https://oasis-open.github.io/cti-documentation/stix/walkthrough
15. State Service for Special Communications and Information Protection of Ukraine. (2023). *On approval of methodological recommendations for responding by cybersecurity subjects to various types of events in cyberspace (Order No. 570)*. https://zakon.rada.gov.ua/rada/show/v0570519-23#Text
16. FIRST. (n.d.). *Traffic light protocol*. https://www.first.org/tlp/
17. OpenAI. (n.d.). *Structured model outputs*.
18. Leto, A., Aguerrebere, C., & Bhati, I. (2024). Toward optimal search and retrieval for RAG (arXiv:2411.07396) [Preprint]. *NeurIPS 2024 Workshop*. https://doi.org/10.48550/arXiv.2411.07396
19. Chornyi, A., & Stopochkina, I. (2025). Graph-based analysis of information flows in Telegram for cybersecurity threat detection. *Cybersecurity: Education, Science, Technique*, 3(27), 368-380. https://doi.org/10.28925/2663-4023.2025.27.746

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (33), 2026
ISSN 2663 – 4023

Андреєв Данило Юрійович
Магістр,
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0009-0009-7908-5388
dandre-ipt24@lll.kpi.ua

Чорний Анатолій Юрійович
Здобувач PhD,
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0009-0001-4147-9084
anacho-ipt23@lll.kpi.ua

Стьопочкіна Ірина Валеріївна
Кандидат технічних наук, доцент, доцент кафедри інформаційної безпеки
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0000-0002-0346-0390
i.stopochkina@kpi.ua

Ільїн Микола Іванович
Кандидат технічних наук, доцент, доцент кафедри інформаційної безпеки
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: 0000-0002-1065-6500
m.ilin@kpi.ua

МЕТОД АВТОМАТИЗАЦІЇ ЗВІТІВ ПРО КІБЕРІНЦИДЕНТИ З ВИКОРИСТАННЯМ LLM

Анотація. Роботу присвячено питанням автоматизації звітності в складі процесів Threat Inteligence. Метою роботи є розробка методу, яка дозволяє зменшити навантаження на працівників, які обробляють та документально фіксують результати кіберінцидентів у відповідності до вимог нормативних документів. Серед основних результатів роботи запропоновано шаблон інструкцій багаторазового використання для великої мовної моделі (ВММ). Представлений шаблон дає змогу надати чіткі вказівки, а саме необхідні та опціональні поля, припустимі значення, які вносяться до полів звіту. Запропоновано програмні моделі на основі бібліотеки Pydantic для генерації та перевірки відповіді у форматі JSON від ВММ. Це дозволяє скоротити довжину інструкцій для ВММ приблизно в 3 рази. Запропоновано архітектуру RAG-пайплайну для врахування конкретного контексту нормативних документів в області звітності щодо кіберінцидентів. Такий пайплайн дозволяє слідувати вимогам законодавства та стандартів без потреби прописувати ці вимоги вручну в інструкціях, що прискорює процес генерації та покращує якість звітів. Розроблено програмну модель, яка дозволяє автоматизовану генерацію звіту з кіберінцидентів. Така модель не потребує ручного заповнення характеристик інциденту, взаємодії користувача з платформою Threat Intelligence на прикладі MISP (Malware Information Sharing Platform). Цей підхід дозволяє знизити час створення звіту з годин до хвилин, і покращити ефективність обміну даними про загрози, уникаючи часових та фінансових вкладень. Ще одним результатом роботи є порівняльний аналіз генерації звітів при використанні різних ВММ, зокрема Claude Sonnet 4.5, Gemini 2.5 pro, Grok xAI, GPT 5, DeepSeek, Llama в розрізі якості та вартості генерації звіту. Для порівняння запропоновано критерії якості звіту, оцінка відповідності критеріям проводилась експертним методом. В результаті виділено моделі Claude Sonnet 4.5, Gemini 2.5 pro як лідерів стосовно якості згенерованих звітів. Встановлено, що ВММ є перспективним інструментом для впровадження в процеси обробки та комунікації в області інцидентів кібербезпеки, їх використання дозволяє повністю автоматизувати процес звітності Threat Intelligence в організації.

Ключові слова: threat intelligence; великі мовні моделі; інциденти кібербезпеки; звітність.

СПИСОК ВИКОРИСТАННИХ ДЖЕРЕЛ

1. Ibrahim, I. M., Soliman, M., & Ossama, S. (2025). Leveraging large language models for document analysis and decision-making in AI chatbots. *Advanced Sciences and Technology Journal*, 2(1), Article 1034. https://doi.org/10.21608/astj.2025.342484.1034
2. Voitsekhovskyi, A., Stopochkina, I., Sun, P., Xie, J., Ilin, M., & Novikov, O. (2026). Detection of vulnerabilities in software for unmanned aerial vehicles by using large language models. *Eastern-European Journal of Enterprise Technologies*, 1(2), 36-47. https://doi.org/10.15587/1729-4061.2026.352029
3. Fezari, M., & Al Dahoud, A. (2026). The evolution of retrieval-augmented generation (RAG) in AI [Preprint]. https://doi.org/10.13140/RG.2.2.27107.62245
4. 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. (2022). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2555
5. Cyber Solidarity Act. (2024). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32025R0038
6. Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). *Computer security incident handling guide (NIST Special Publication 800-61 Rev. 2)*. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-61r2
7. Ainslie, S., Thompson, D., Maynard, S., & Ahmad, A. (2023). Cyber-threat intelligence for security decision-making: A review and research agenda for practice. *Computers & Security*, 132, 103352. https://doi.org/10.1016/j.cose.2023.103352
8. Lin, X., et al. (2025). IRCopilot: Automated incident response with large language models (arXiv:2505.20945) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2505.20945
9. Novikov, O., Ilin, M., Stopochkina, I., Ovcharuk, M., & Voitsekhovskyi, A. (2025). Application of LLM in UAV route planning tasks to prevent data exchange availability violations. *Cybersecurity: Education, Science, Technique*, 1(29), 420–431. https://doi.org/10.28925/2663-4023.2025.29.892
10. Sohi, S., Balan, D., Anjomshoaa, A., & Polleres, A. (2024). Towards harmonised rail safety knowledge: LLM techniques for EU accident report processing. In *CEUR Workshop Proceedings*. https://ceur-ws.org/Vol-4079/short4.pdf
11. MISP Project. (2025). *Features of MISP, the open source threat sharing platform*. Retrieved October 30, 2025, from https://www.misp-project.org/features/
12. Dulaunoy, A., & Iklody, A. (n.d.). *MISP core format*. MISP Standard. https://www.misp-standard.org/rfc/misp-standard-core.html
13. Rutkowski, A., Kadobayashi, Y., & Furey, I. (2010). CYBEX: The cybersecurity information exchange framework. *ACM SIGCOMM Computer Communication Review*, 40(5).
14. OASIS Open. (n.d.). *STIX introductory walkthrough*. https://oasis-open.github.io/cti-documentation/stix/walkthrough
15. State Service for Special Communications and Information Protection of Ukraine. (2023). *On approval of methodological recommendations for responding by cybersecurity subjects to various types of events in cyberspace (Order No. 570)*. https://zakon.rada.gov.ua/rada/show/v0570519-23#Text
16. FIRST. (n.d.). *Traffic light protocol*. https://www.first.org/tlp/
17. OpenAI. (n.d.). *Structured model outputs*.
18. Leto, A., Aguerrebere, C., & Bhati, I. (2024). Toward optimal search and retrieval for RAG (arXiv:2411.07396) [Preprint]. *NeurIPS 2024 Workshop*. https://doi.org/10.48550/arXiv.2411.07396
19. Chornyi, A., & Stopochkina, I. (2025). Graph-based analysis of information flows in Telegram for cybersecurity threat detection. *Cybersecurity: Education, Science, Technique*, 3(27), 368-380. https://doi.org/10.28925/2663-4023.2025.27.746

Отримано редакцією журналу / Received: 10.02.26
Прорецензовано / Revised: 25.02.26
Схвалено до друку / Accepted: 25.06.26