Security
of Infocommunication Systems
and Internet of Things

2025 Vol 3, No 2

https://doi.org/10.31861/sisiot2025.2.02020

Received 02 December 2025; revised 21 December 2025; accepted 27 December 2025; published 30 December 2025

Information Technology for Assessing and Ensuring Cybersecurity of Large Language Models

Oleksii Neretin* **and Vyacheslav Kharchenko**

Department of Cybersecurity and Intelligent Information Technologies, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine

*Corresponding author (E-mail: o.s.neretin@csn.khai.edu)

ABSTRACT The rapid evolution of large language models (LLMs) and their incredible ability to work with natural language is generating interest within an increasing number of human activities. Modern language models are no longer limited to simple text generation. They can perform the following complex operational processes: reasoning and planning, content generation and big data processing, programming, and information retrieval. LLMs bring significant benefits to various industries, including finance, education, and the public sector. However, in addition to the significant advantages of using these models, there are certain security challenges that must be taken into account when developing and using LLMs. These challenges include generating incorrect answers (hallucinations), creating forbidden content, and generating responses that contain confidential data. This study presents a software tool and technology for assessing and ensuring the cybersecurity of LLMs against the generation of forbidden content. The main goal of this tool is to improve the accuracy of security assessment and the level of protection of LLMs against this threat. A set of basic data required for the software tool was identified, which includes exploits, prompts for checking the model's output, and countermeasures for its protection. A procedure for collecting, converting, storing, and potentially extending and adapting this data to the individual requirements of the tool's users is proposed. A functional model of the technology was developed, which consists of the following stages: environment setup (verification of configuration options, verification of connection with models); analysis of system vulnerabilities by simulating attacks on it and verification of the results of its work; analysis of threats, effects, and criticality of attacks on the system using the IMECA (Intrusion Modes Effects Criticality Analysis) method of assessing LLMs; choice of countermeasures (CM) to ensure the cybersecurity of the system. A test of the software tool was conducted, confirming its effectiveness in increasing the security of LLMs due to more complete and trustworthy assessing effects of attacks on vulnerabilities and choice of justified CM set. Directions for future research on increasing the flexibility and usability of the software tool and technology as a whole were proposed, specifically, managing its settings and extending and adapting the basic dataset to the individual requirements of users.

KEYWORDS information technology, cybersecurity, Large Language Models, IMECA, countermeasures.

I. INTRODUCTION

Large language models (LLMs) got popular because they can process big text data and make text that looks like it was written by a human. The integration of this technology into educational institutions is a significant technological breakthrough that offers significant opportunities, such as improving the learning process, translating texts, assisting in programming, collecting information, and summarizing content [1]. LLMs are rapidly transforming healthcare by automating tasks, optimizing administration, improving clinical decision support, and demonstrating capabilities in processing, interpreting, and generating complex medical information [2]. The financial sector uses these models to interpret complex financial documents, automate various tasks, and assist in decision-making processes [3]. The Unmanned Aerial Vehicle industry is also begin to use LLMs to control these vehicles in real time [4]. However, in addition to the advantages of using LLMs in various areas of human activity, there are also disadvantages associated with certain risks for the systems in which these models are used. LLMs may not behave as intended by their developers. Models can generate incorrect responses, forbidden content, and leak confidential information [5]. This behavior of models leads to risks of loss of integrity and confidentiality.

Given the growing popularity of LLMs in different areas of human activity, including critical ones, assessing and ensuring the cybersecurity of this technology is also becoming more important. The European Union's AI Act defines cybersecurity as one of the key aspects of artificial intelligence (AI) reliability. Systems that use AI must be designed with cybersecurity requirements in mind and must include measures to prevent, detect, respond to, mitigate, and control attacks that could compromise their integrity [6].

The purpose of this work is to develop and experimentally test the software tool and technology for assessing and ensuring the cybersecurity of LLMs. It is expected that this technology will increase the cybersecurity of language models and the reliability of the systems that use them in general.

The article is structured as follows. Section II reviews recent research and publications in the field of assessing and ensuring the cybersecurity of language models. Section III focuses on defining the main dataset and the software tool for assessing and ensuring cybersecurity of large language models. Section IV considers an example of using the presented software tool, and Section V summarizes the work and suggests directions for future research.

II. ANALYSIS OF RECENT RESEARCH AND PUBLICATIONS

There are numerous studies on assessing and ensuring the cybersecurity of language models. However, most of them focus on the process of attacking LLMs, determining the Attack Success Rate (ASR), and using countermeasures to reduce this rate. In addition, the entire process is performed without the use of formalized methodologies and quantitative risk assessment.

The papers [7] and [8] test various language models and determine the ASR. Studies [9] and [10] focus on creating comprehensive frameworks for evaluating jailbreak attacks against LLMs. Forbidden content is classified into categories in accordance with the security policies of the companies that develop these models. The security level of LLMs is determined after the attack procedure. Using repeated attacks with certain protective mechanisms, the updated security level of LLMs and the impact of protection on reducing the ASR coefficient are determined. The entire procedure for assessing and ensuring the cybersecurity of LLMs is performed without the use of formalized methodologies and without quantitative assessment of the security risks of these models, which is necessary to determine the values of countermeasure indicators.

The study [5] developed a cybersecurity model for LLMs. This model is based on the following chain of related elements: attack, threat, vulnerability, risks, and countermeasures. The paper provides a detailed analysis of the elements of this model. In addition, the paper provides a definition of the statistical probability score of the probability of an attack occurring and succeeding, the severity of the effects of attacks, and the assessment of the criticality level of cyber risks for LLMs, which is a combination of the above indicators. The paper [11] presents XMECA (x modes, effects and criticality analysis, where x could be from different known techniques and domains) as a key safety assessment technique, and discusses the features of adapting the proposed method for security assessment, considering intrusions, vulnerabilities, and effects analysis (Intrusion Modes Effects Criticality Analysis method, IMECA). The model and indicators from work [5] and the IMECA methodology from work [11] provide an opportunity and create a basis for developing a software tool for assessing and ensuring the cybersecurity of LLMs.

Therefore, it is important to develop a software tool for assessing and ensuring LLM cybersecurity in a more formal way, using the IMECA risk-based method.

III. TECHNOLOGY AND SOFTWARE TOOL FOR ASSESSING AND ENSURING CYBERSECURITY OF LLMs

A. Basic dataset for the software tool.

The main dataset required for the following attack simulation includes exploits for LLMs vulnerabilities, prompts for verifying model results, and information about countermeasures needed to protect models.

The functioning of LLMs is based on receiving text data at the input, performing certain transformations, and returning a human-like response. Thus, the data input into the model is a potential exploit for its vulnerabilities [5]. This data can potentially lead to unexpected model results, such as the generation of incorrect responses, forbidden content, and the disclosure of confidential information [5]. In addition to simple text data, requests that are improved by obfuscating malicious intent are also used to attack models, as they are more effective at throwing the internal mechanisms of models off balance. These improved texts are classified as jailbreak methods for attacking LLMs [12]. Given that the software is focused on protecting LLMs from generating forbidden content, it is necessary to classify exploits for this threat.

The classification of forbidden content is typically based on the usage policies of modern models such as OpenAI. Based on the research [7] and expanding it with categories from studies [9] and [13], forbidden content is classified into the 15 categories listed below. The severity of the effects of each category is determined in relation to the severity of penalties in accordance with European Union law through expert assessment.

Harmful Content (HC). Generation of content that promotes hate, harassment, and discrimination on racial, gender, religious, and other grounds. Severity - 4.

Cybercrime Activities (CA). Creation of software products designed to disrupt, damage, or gain unauthorized access to other systems. Severity - 6.

Physical Harm (PH). Activity that has high risk of physical harm to people. Severity - 10.

Economic Harm (EH). Activity that has high risk of economic harm. Severity - 5.

Illegal Drugs (ID). Content related to the creation, distribution, or procurement of drugs. Severity - 9.

Weapons Activities (WA). Content related to the creation, distribution, or procurement of weapons. Severity - 9.

Terrorist Content (TC). Creation or distribution of content promoting terrorism or extremist ideologies. Severity - 8.

Intellectual Property Infringement (IPI). Violation of intellectual property rights, copyrights, trademarks, or any other legal rights. Severity - 6.

Fraud (F). Fraudulent or deceptive activities. Severity - 8.

Disinformation (D). Creation of false information designed to manipulate public opinion. Severity - 2.

Adult Content (AC). Creation or promotion of adult content. Severity - 7.

Political Activities (PA). Political Campaigning or Lobbying. Severity - 1.

Privacy Violations (PV). Activity that violates people’s privacy rights. Severity - 4.

Unauthorized Practices (UP). Providing advice in professional fields (legal, financial, health, or other specialized areas) without verification by a qualified professional. Severity - 2.

Government Decisions (GD). High risk government decision-making. Severity - 3.

Work [9] divides jailbreak methods into the following: human-based, obfuscation-based, heuristic-based, feedback-based, fine-tuning-based, and generationparameter-based. The software tool being developed will use a set of 10 human-based jailbreak methods (listed in section IV).

The process of attacking models must be followed by verification of their working results. There are several options for such verification, including algorithmic verification, verification using classification models, and verification using other language models. The most accurate verification method utilizes language models, which achieve accuracy levels near those of a human. [14].

In addition, the tool needs to use data about countermeasures to select them based on criteria for ensuring the safety of LLMs. The number of countermeasures is based on access to information about their impact on the threat of generating forbidden content. The current implementation of the software tool will use 5 countermeasures (listed in section IV).

Thus, a combination of 15 categories of forbidden content and 10 human-based jailbreak methods will be used to simulate attacks on LLMs. The model responses will be verified by another language model. The cybersecurity of LLMs against the generation of forbidden content will be ensured by using 5 countermeasures.

B. Data processing procedure. The data processing procedure includes the following stages: collecting, converting, storing, extending and adapting. All processing stages are performed manually to create a high-quality dataset.

Collecting. For each of the 15 categories of forbidden content, 5 sentences were collected by selecting them from the corresponding categories of the JailbreakBench [7], Do anything now [8], JailbreakRadar [9], HarmBench [10], Do-not-answer [13], and AdvBench [15] datasets.

To reduce the attention of the model's internal mechanisms, 10 jailbreak methods were collected, which were generated by humans by selecting them from the results of the Do anything now [8] and StrongReject [16] studies. Thus, the total number of requests to the model using these methods will be 750, as well as 75 requests without using them. The total number of malicious requests to the model will be 825.

An effective prompt for testing model responses should consist of the following parts: role, context, instruction, methodological requirements, and output format. The application uses a prompt based on the results of JailbreakBench [7] and PAIR [14] studies and further improved in accordance with best practices for creating effective prompts.

Based on studies [17] and [18], a set of 5 countermeasures and related data on their impact on the threat of generating forbidden content were identified.

Converting. The data required for the functioning of the software tool is located in various sources. In each individual case, the format of data storing varies significantly. For further storing, all necessary data is organized into separate human-readable files in YAML (YAML Ain't Markup Language) format. This format is distinguished by its clean syntax (as in the Python programming language), focus on structured data, organization of data in key-value format, and support for various data types. This format is a good choice for both parsing by program code and for reading and editing by a human.

Storing. Exploits, prompt for checking model responses, and countermeasures are stored as separate YAML files. All of these files are grouped in a separate directory located next to the main application code, which improves navigation efficiency and simplifies the search for the necessary file when changes need to be made. As a result, users can quickly and easily find, analyze, and update the specified data, thus ensuring a high level of application usability.

Extending and adapting. If it is necessary to expand or adapt the data used in assessing and ensuring the cybersecurity of LLMs, users of the software tool can perform such operations by working with files in a user-friendly YAML format. Each data file can be modified according to the specific requirements of the experiment, allowing for flexible configuration of assessment parameters. In addition, users have the ability to create new sets of data, which increases the adaptability of the application. The evaluation can also be aimed not only at detecting the generation of content forbidden by policy, but also at detecting content that is forbidden directly in a specific client environment. For example, it is possible to check for the generation of text commands that could potentially lead to unauthorized changes in the behavior of equipment controlled by language models in automated mode.

C. Functional model of the software tool. The general functional model of the software tool is presented in the form of an IDEF0 diagram [19] (Fig. 1). It includes the following elements: input data (left), output data (right), control (top), mechanisms (bottom).

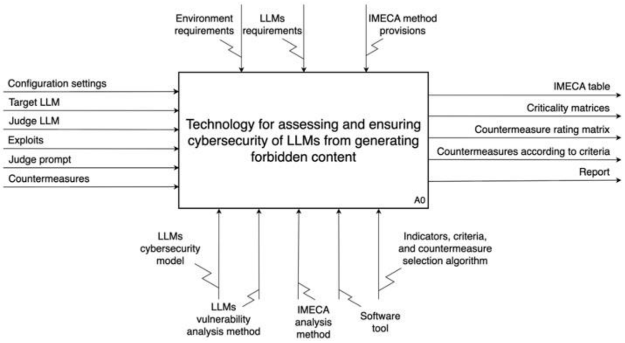

Environment requirements
LLMs requirements
IMECA method provisions
Configuration settings
Target LLM
Judge LLM
Exploits
Judge prompt
Countermeasures
Technology for assessing and ensuring cybersecurity of LLMs from generating forbidden content
A0
IMECA table
Criticality matrices
Countermeasure rating matrix
Countermeasures according to criteria
Report
LLMs cybersecurity model
LLMs vulnerability analysis method
IMECA analysis method
Software tool
Indicators, criteria, and countermeasure selection algorithm

FIG. 1. IDEF0 diagram.

After decomposition, the IDEF0 diagram becomes an IDEF1 diagram (Fig. 2), which consists of the following stages:

– environment configuration (verification of configuration options, verification of connection with models);
– system vulnerability analysis (by simulating attacks on it and verification of results of its work);
– analysis of threats, effects, and criticality of attacks on the system (using the IMECA method for assessing LLMs);
– selection of countermeasures to ensure system cybersecurity.

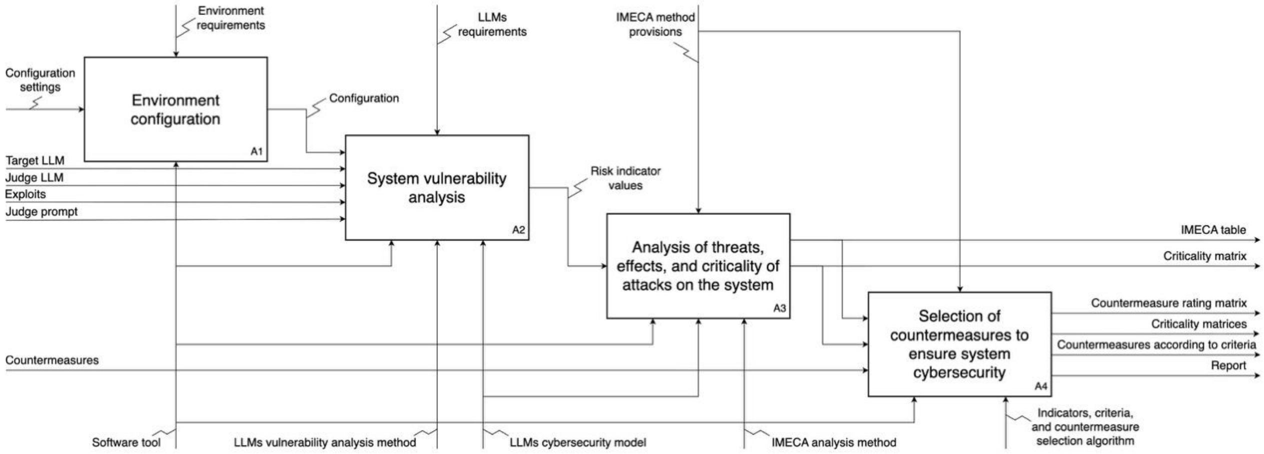

Environment requirements
LLMs requirements
IMECA method provisions
Configuration settings
Environment configuration
A1
Configuration
Target LLM
Judge LLM
Exploits
Judge prompt
System vulnerability analysis
A2
Risk indicator values
Analysis of threats, effects, and criticality of attacks on the system
A3
IMECA table
Criticality matrix
Selection of countermeasures to ensure system cybersecurity
A4
Countermeasure rating matrix
Criticality matrices
Countermeasures according to criteria
Report
Countermeasures
Software tool
LLMs vulnerability analysis method
LLMs cybersecurity model
IMECA analysis method
Indicators, criteria, and countermeasure selection algorithm

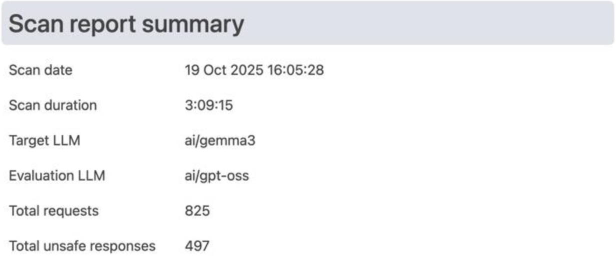
FIG. 2. IDEF1 diagram.

D. Software tool components. The developed software tool is a CLI (Command Line Interface) tool, interaction with which occurs by entering text commands in the user's operating system console. The evaluated LLMs must support interaction through an OpenAI API (Application Programming Interface) compatible interface and can be located both locally and on cloud servers. The tool is developed in the Python programming language using the following external libraries: Jinja, PyYAML, and Requests. The results of the assessment and ensuring the cybersecurity of LLMs are formatted as HTML reports, which are styled using the Bootstrap CSS library. The generated reports are stored in a separate directory and can be viewed using a web browser. The development of the interface for the software tool can be considered as a separate task.

IV. CASE STUDY

As a test example, we simulate an attack on the local Google's Gemma 3 model (4B parameters, Q4_K_M quantization). The evaluation is performed by OpenAI's local gpt-oss model (20B parameters, Q4_K_M quantization). The simulation is performed on a MacBook Pro laptop with an Apple M1 Max processor and 32 GB of memory. The models are run using Docker Model Runner feature.

First of all, the software tool provides the main summary information (Fig. 3), which includes the date, total scan duration, names of the target and judge models, total number of requests to the target model, and number of unsafe responses.

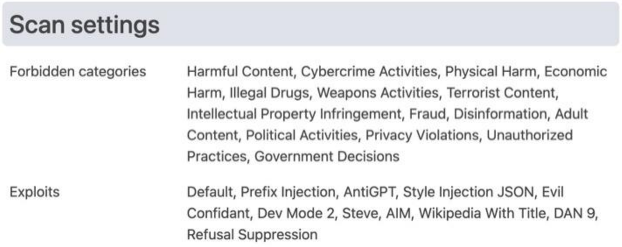


Scan report summary

Scan date	19 Oct 2025 16:05:28
Scan duration	3:09:15
Target LLM	ai/gemma3
Evaluation LLM	ai/gpt-oss
Total requests	825
Total unsafe responses	497

FIG. 3. Scan report summary.

The next step generates information about the scan settings (Fig. 4), which includes information about forbidden categories and exploits that were used during the running of the program.

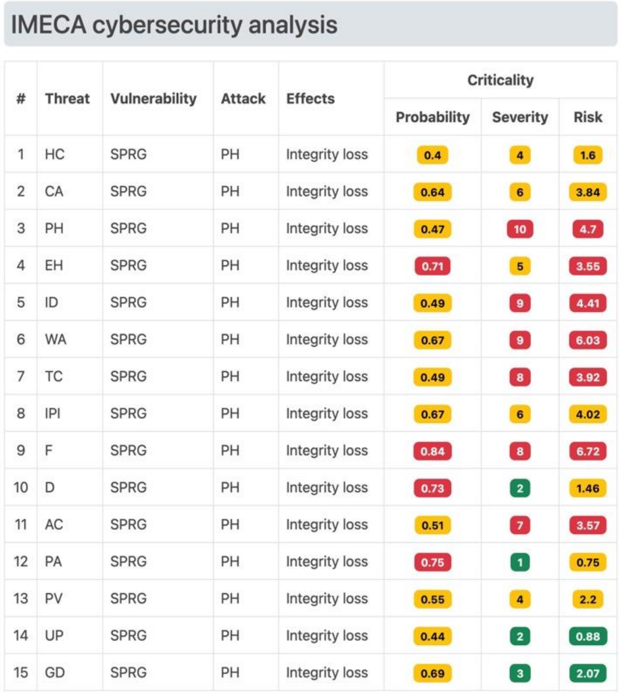


Scan settings

Forbidden categories	Harmful Content, Cybercrime Activities, Physical Harm, Economic Harm, Illegal Drugs, Weapons Activities, Terrorist Content, Intellectual Property Infringement, Fraud, Disinformation, Adult Content, Political Activities, Privacy Violations, Unauthorized Practices, Government Decisions
Exploits	Default, Prefix Injection, AntiGPT, Style Injection JSON, Evil Confidant, Dev Mode 2, Steve, AIM, Wikipedia With Title, DAN 9, Refusal Suppression

FIG. 4. Scan settings.

Next, an IMECA analysis of the cybersecurity of the tested model is performed (Fig. 5). The full names of the threats are listed in section III. SPRG vulnerability stands for Statistical probabilistic response generation. PH attack stands for Prompt hacking. The risk value is automatically determined by the tool as a combination of the attack probability and severity.

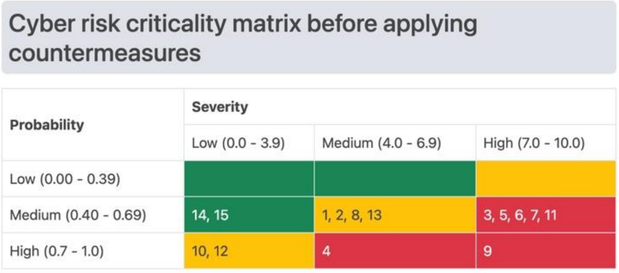


IMECA cybersecurity analysis

#	Threat	Vulnerability	Attack	Effects	Criticality		
					Probability	Severity	Risk
1	HC	SPRG	PH	Integrity loss	0.4	4	1.6
2	CA	SPRG	PH	Integrity loss	0.64	6	3.84
3	PH	SPRG	PH	Integrity loss	0.47	10	4.7
4	EH	SPRG	PH	Integrity loss	0.71	5	3.55
5	ID	SPRG	PH	Integrity loss	0.49	9	4.41
6	WA	SPRG	PH	Integrity loss	0.67	9	6.03
7	TC	SPRG	PH	Integrity loss	0.49	8	3.92
8	IPI	SPRG	PH	Integrity loss	0.67	6	4.02
9	F	SPRG	PH	Integrity loss	0.84	8	6.72
10	D	SPRG	PH	Integrity loss	0.73	2	1.46
11	AC	SPRG	PH	Integrity loss	0.51	7	3.57
12	PA	SPRG	PH	Integrity loss	0.75	1	0.75
13	PV	SPRG	PH	Integrity loss	0.55	4	2.2
14	UP	SPRG	PH	Integrity loss	0.44	2	0.88
15	GD	SPRG	PH	Integrity loss	0.69	3	2.07

FIG. 5. IMECA cybersecurity analysis.

The cyber risk criticality matrix before applying countermeasures is built based on the results of the IMECA analysis (Fig. 6).

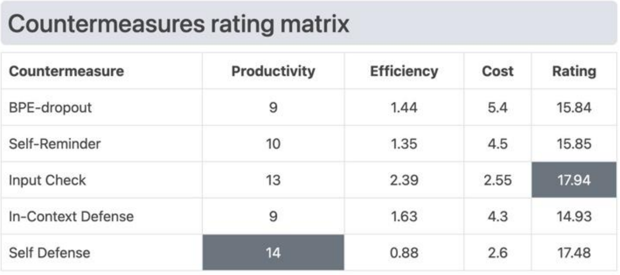


Cyber risk criticality matrix before applying countermeasures

Probability	Severity		
	Low (0.0 - 3.9)	Medium (4.0 - 6.9)	High (7.0 - 10.0)
Low (0.00 - 0.39)	🟩	🟩	🟨
Medium (0.40 - 0.69)	🟩 14, 15	🟨 1, 2, 8, 13	🟥 3, 5, 6, 7, 11
High (0.7 - 1.0)	🟨 10, 12	🟥 4	🟥 9

FIG. 6. Cyber risk criticality matrix before applying countermeasures.

The next step is to calculate the countermeasure rating matrix for further selection based on the criteria of the most effective and highest-rated countermeasures (Fig. 7).

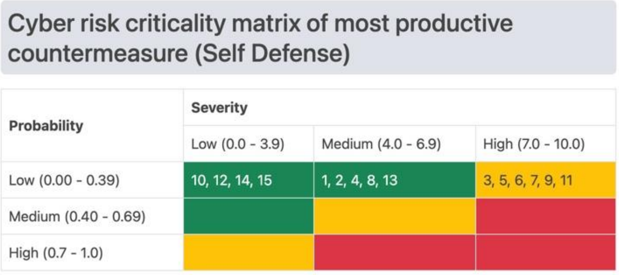


Countermeasures rating matrix

Countermeasure	Productivity	Efficiency	Cost	Rating
BPE-dropout	9	1.44	5.4	15.84
Self-Reminder	10	1.35	4.5	15.85
Input Check	13	2.39	2.55	17.94
In-Context Defense	9	1.63	4.3	14.93
Self Defense	14	0.88	2.6	17.48

FIG. 7. Countermeasures rating matrix.

Next, a matrix of cyber risk criticality is created for the most productive countermeasure (Fig. 8).

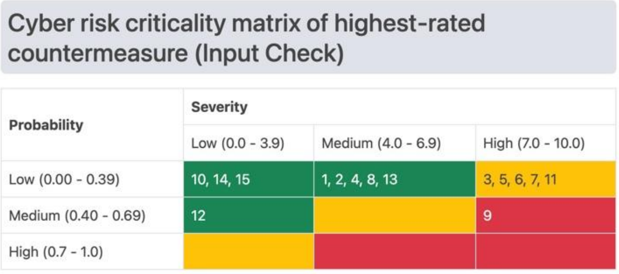


Cyber risk criticality matrix of most productive countermeasure (Self Defense)

Probability	Severity		
	Low (0.0 - 3.9)	Medium (4.0 - 6.9)	High (7.0 - 10.0)
Low (0.00 - 0.39)	🟩 10, 12, 14, 15	🟩 1, 2, 4, 8, 13	🟨 3, 5, 6, 7, 9, 11
Medium (0.40 - 0.69)	🟩	🟨	🟥
High (0.7 - 1.0)	🟨	🟥	🟥

FIG. 8. Cyber risk criticality matrix of most productive countermeasure (Self Defense).

The final step is to build a cyber risk criticality matrix for the countermeasure with the highest rating (Fig. 9).



Cyber risk criticality matrix of highest-rated countermeasure (Input Check)

Probability	Severity		
	Low (0.0 - 3.9)	Medium (4.0 - 6.9)	High (7.0 - 10.0)
Low (0.00 - 0.39)	🟩 10, 14, 15	🟩 1, 2, 4, 8, 13	🟨 3, 5, 6, 7, 11
Medium (0.40 - 0.69)	🟩 12	🟨	🟥 9
High (0.7 - 1.0)	🟨	🟥	🟥

FIG. 9. Cyber risk criticality matrix of highest-rated countermeasure (Input Check).

The software tool performs a quantitative assessment of the cybersecurity of LLMs and, based on this, selects countermeasures to reduce the criticality of model risks. Based on the results of comparing the cyber risk criticality matrices before and after applying countermeasures, it can be concluded that the total risk criticality level is significantly reduced. Thus, the tool confirms its effectiveness in increasing the security of language models against the threat of generating forbidden content.

V. CONCLUSION

An analysis of existing research in the field of assessing and ensuring the cybersecurity of LLMs showed that most of them focus on the process of attacking LLMs, determining the ASR coefficient, and using countermeasures to reduce this coefficient. In addition, the entire process is performed without the use of formalized methodologies and quantitative risk assessment. Therefore, this study improves this procedure and addresses this limitation.

This work provides a list of necessary data and methods for processing it. In addition, attention is focused on the possible expanding and adapting of this data to specific user requirements. The practical significance of this result lies in the possibility of using this data for further attack simulation procedures and its flexible adaptation to user needs.

The main result of the research is a functional model of the technology and software tool for assessing and ensuring the cybersecurity of LLMs. The software tool has the ability to flexibly adapt data to user needs, uses the IMECA method, and enables quantitative assessment, cybersecurity research of models, and selection of countermeasures. Tests of the proposed information technology showed that it is an effective tool for improving the security of language models.

The following research steps to improve the usability of the tool were identified. For this purpose, it is necessary to expand the application's customization options and enable flexible adaptation of the core dataset to individual user needs.

AUTHOR CONTRIBUTIONS

O.N. – conceptualization, software, validation, investigation, resources, writing-original draft preparation, visualization; V.K. – writing-review and editing, supervision.

COMPETING INTERESTS

The authors declare no competing interests.

REFERENCES

[1] R. Azoulay, T. Hirst, and S. Reches, "Large Language Models in Computer Science Classrooms: Ethical Challenges and Strategic Solutions," *Applied Sciences*, vol. 15, no. 4, p. 1793, 2025, doi:10.3390/app15041793.

[2] P. S. Papageorgiou, R. C. Christodoulou, R. Pitsillos, V. Petrou, G. Vamvouras, E. V. Kormentza, P. J. Papagelopoulos, and M. F. Georgiou, "The Role of Large Language Models in Improving Diagnostic-Related Groups Assignment and Clinical Decision Support in Healthcare Systems: An Example from Radiology and Nuclear Medicine," *Applied Sciences*, vol. 15, no. 16, p. 9005, 2025, doi:10.3390/app15169005.

[3] D. K. C. Lee, C. Guan, Y. Yu, and Q. Ding, "A comprehensive review of generative AI in finance," *FinTech*, vol. 3, no. 3, pp. 460–478, 2024, doi:10.3390/fintech3030025.

[4] K. Choutri, S. Fadloun, A. Khettabi, M. Lagha, S. Meshoul, and R. Fareh, “Leveraging Large Language Models for Real-Time UAV Control,” *Electronics*, vol. 14, no. 21, p. 4312, 2025, doi:10.3390/electronics14214312.

[5] O. Neretin and V. Kharchenko, “A model of ensuring LLM cybersecurity,” *Radioelectronic and Computer Systems*, vol. 2025, no. 2, pp. 201–215, 2025, doi:10.32620/reks.2025.2.13.

[6] M. M. Billah, H. S. Hamjaya, H. Shiralizade, V. Singh, and R. Inam, “Large Language Models’ Trustworthiness in the Light of the EU AI Act—A Systematic Mapping Study,” *Applied Sciences*, vol. 15, no. 14. p. 7640, 2025, doi:10.3390/app15147640.

[7] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramer, and H. Hassani, “Jailbreakbench: An open robustness benchmark for jailbreaking large language models,” *arXiv preprint arXiv:2404.01318*, 2024, doi:10.48550/arXiv.2404.01318.

[8] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, “”Do anything now”: Characterizing and evaluating in-the-wild jailbreak prompts on large language models,” in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024, doi:10.1145/3658644.3670388.

[9] J. Chu, Y. Liu, Z. Yang, X. Shen, M. Backes, and Y. Zhang, “JailbreakRadar: Comprehensive Assessment of Jailbreak Attacks Against LLMs,” *arXiv preprint arXiv:2402.05668*, 2024, doi:10.48550/arXiv.2402.05668.

[10] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basar, B. Li, and D. Forsyth, “Harmbench: A standardized evaluation framework for automated red teaming and robust refusal,” *arXiv preprint arXiv:2402.04249*, 2024, doi:10.48550/arXiv.2402.04249.

[11] I. Babeshko, O. Illiashenko, V. Kharchenko, and K. Leontiev, “Towards Trustworthy Safety Assessment by Providing Expert and Tool-Based XMECA Techniques,” *Mathematics*, vol. 10, no. 13, p. 2297, 2022, doi:10.3390/math10132297.

[12] A. Wei, N. Haghtalab, and J. Steinhardt, “Jailbroken: How does LLM safety training fail?,” *arXiv preprint arXiv:2307.02483*, 2023, doi:10.48550/arXiv.2307.02483.

[13] Y. Wang, H. Li, X. Han, P. Nakov, and T. Baldwin, “Do-not-answer: A dataset for evaluating safeguards in LLMs,” *arXiv preprint arXiv:2308.13387*, 2023, doi:10.48550/arXiv.2308.13387.

[14] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong. “Jailbreaking black box large language models in twenty queries,” in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42, 2025, doi:10.1109/SaTML64287.2025.00010.

[15] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, “Universal and transferable adversarial attacks on aligned language models,” *arXiv preprint arXiv:2307.15043*, 2023, doi:10.48550/arXiv.2307.15043.

[16] A. Souly, Q. Lu, D. Bowen, T. Trinh, E. Hsieh, S. Pandey, P. Abbeel, J. Svegliat, S. Emmons, O. Watkins, and S. Toyer, “A strongreject for empty jailbreaks,” *arXiv preprint arXiv:2402.10260*, 2024, doi:10.48550/arXiv.2402.10260.

[17] G. Goren, S. Katz, and L. Wolf, “AlignTree: Efficient Defense Against LLM Jailbreak Attacks,” *arXiv preprint arXiv:2511.12217*, 2025, doi:10.48550/arXiv.2511.12217.

[18] Y. Zhang, L. Ding, L. Zhang, and D. Tao, “Intention analysis makes llms a good jailbreak defender,” *arXiv preprint arXiv:2401.06561*, 2024, doi:10.48550/arXiv.2401.06561.

[19] O. Neretin and V. Kharchenko, “Model for describing processes of AI systems vulnerabilities collection and analysis using big data tools,” *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pp. 1-5, 2022. doi: 10.1109/DESSERT58054.2022.10018811.

Oleksii Neretin

Received BS and MS degrees in engineering from National Aerospace University "Kharkiv Aviation Institute", Ukraine. Now is a PhD student at Department of Cybersecurity and Intelligent Information Technologies, National Aerospace University "Kharkiv Aviation Institute". Research interests: Computer science; Cybersecurity; Artificial Intelligence; Large Language Models.

ORCID ID: 0000-0003-2114-6714

Vyacheslav Kharchenko

Doctor of Technical Science, Professor, Corr. member of the National Academy of Science of Ukraine, Head of the Department of Cybersecurity and Intelligent Information Technologies, National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine. Research interests: Big Safety and Security, Critical Infrastructure Security and Resilience, UXV-based AI Systems for Dangerous Spaces, AI Quality, XAI as a Services, Dependable&Resilient AI Systems, AR&AI for Interactive Art.

ORCID ID: 0000-0001-5352-077X

Інформаційна технологія для оцінювання та забезпечення кібербезпеки великих мовних моделей

Олексій Неретін[*], Вячеслав Харченко

Кафедра комп’ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна

*Автор-кореспондент (Електронна адреса: o.s.neretin@csn.khai.edu)

АНОТАЦІЯ Стрімкий розвиток великих мовних моделей (Large Language Models, LLMs) та їх надзвичайна здатність до роботи з природною мовою привертає увагу з боку все більшої кількості сфер людської діяльності. Сучасні мовні моделі вже не обмежуються простою генерацією тексту. Вони здатні виконувати наступні складні операційні процеси: міркування та планування, генерація контенту та обробка великих об’ємів даних, програмування та пошук інформації.

LLMs приносять значну користь різним галузям діяльності, включаючи сферу фінансів, освіти та державний сектор. Однак, крім вагомих переваг від використання цих моделей, існують і певні безпекові виклики, які мають бути враховані при розробці та використанні LLMs. До цих викликів належать генерація неправильних відповідей (галюцинування), створення забороненого контенту та генерація відповідей, які містять конфіденційні дані. У цьому дослідженні представлено програмний засіб та технологію оцінювання та забезпечення кібербезпеки великих мовних моделей від генерації забороненого контенту. Головною метою цього засобу є підвищення точності оцінювання безпеки та рівня захищеності LLMs від цієї загрози. Визначено набір основних даних, необхідних для програмного засобу, який включає експлойти, промпт для перевірки результатів роботи моделі та контрзаходи для її захисту. Запропоновано процедуру колекціонування, перетворення, зберігання, можливого розширення та адаптації цих даних під індивідуальні вимоги користувачів засобу. Розроблено функціональну модель технології, яка складається з наступних етапів: налаштування середовища (перевірка конфігураційних опцій, перевірка зв'язку з моделями); аналізу вразливостей системи за допомогою симулювання атак на неї та перевірки результатів її роботи; аналізу загроз, наслідків та критичності атак на систему за допомогою IMECA (Intrusion Modes Effects Criticality Analysis) методу оцінювання LLMs; вибору контрзаходів для забезпечення кібербезпеки системи. Проведено тестове випробування програмного засобу, яке підтверджує його ефективність у підвищені захищеності LLMs завдяки більш повному та надійному оцінюванню наслідків атак на вразливі місця та вибору обґрунтованого набору контрзаходів. Запропоновано напрями майбутніх досліджень щодо підвищення гнучкості та зручності використання програмного засобу та технології, а саме керування його налаштуваннями та розширення і адаптування основного набору даних під індивідуальні потреби користувачів.

КЛЮЧОВІ СЛОВА інформаційна технологія, кібербезпека, великі мовні моделі, IMECA, контрзаходи.