ISSN - 2456-1142

International journal of humanities and information technology Volume 7, Issue 1, 2025

AI-Based Threat Modeling for Secure Software Design

Nishad Kane[1] , Dhvani Kansara[2]

[1]Arizona State University, USA

[2]University of Southern California, USA

Abstract

The introduction of Artificial Intelligence (AI) software design is a revolution in cybersecurity. This paper discusses how AI algorithms can be used to predict the attack surface of a software application at the design stage and thus improve vulnerabilities at the initial stage of the software development life cycle. The analysis of AI potential to detect patterns and anomalies that could be otherwise missed using traditional security methods is conducted using machine learning models. These techniques involve using some AI methods, including supervised and unsupervised learning, on historical software data to predict outliers. The findings indicate that AI-based threat modeling can be an effective approach to predicting vulnerabilities, and it provides more precise and timely information than traditional techniques. Finally, the research points out the significance of constructive countermeasures in the development lifecycle, and how AI may lead to a more secure software design by predicting threats prior to their occurrence. The evidence highlights how AI can be critical in the future of secure software engineering.

Keywords: Threat Modeling, Artificial Intelligence, Software Design, Vulnerability Detection, Machine Learning, Active Security, Computer Security, Predictive Algorithms, Software Vulnerability, Secure Design

International journal of humanities and information technology (2025)

INTRODUCTION

Background to the Study

One of the most significant concepts in the development of the modern systems is secure software design when the cyber threats become more advanced and complicated. With the increasing integration of software applications in all areas of daily life, including the medical field and accounting, their security comes as a top priority to prevent data breaches, system failures, and cyberspace intrusions. The dynamic character of these threats, such as zero-day vulnerabilities, ransomware and advanced persistent threats (APT), is requiring proactive security practices over reactive ones. Although effective to some degree, traditional security practices are generally unable to respond to vulnerabilities early in the development process. Artificial intelligence (AI) is becoming a potent means of automating the detection and prevention of these threats. The threat modelling based on AI will assist in the prediction of the possible vulnerabilities at the design stage to respond faster and more efficiently and to eradicate the risk of falling victim to a security attack (Sun et al., 2023).

Overview

AI-based threat modeling can be defined as the use of artificial intelligence methods (machine learning, neural networks, etc.) to anticipate the existence of vulnerabilities in software at the early stage of development. This proactive approach is beneficial in identifying potential threats at the initial level as the developer can react to them instantly rather than them being identified at a later development point. Using AI, developers will be able to process a large number of data sets and detect patterns and possible attack vectors more precisely than manually. AI, especially when embedded into software security frameworks, is used to improve threat detection and mitigation. It automates complex, time-consuming tasks, streamlines them, and provides real-time decision-oriented information. Another option is to implement AI within the software development lifecycle, which can allow designers to develop safer and stronger systems that learn fewer vulnerabilities that might lead to serious consequences (Bhuyan et al., 2020).

Problem Statement

The common security practices within the software development industry have been to identify the attack

Corresponding Author: Nishad Kane, Arizona State University, USA.

How to cite this article: Kane, N. Dhvani Kansara2 (2025). AI-Based Threat Modeling for Secure Software Design. *International journal of humanities and information technology* 7(1), 32-38.

Source of support: Nil

Conflict of interest: None

surfaces too late into the development process and expose systems to possible exploitations. The security testing and analysis approaches used in the past cannot keep up with the new threats as software systems get more complex. The fast development of new attack vectors needs smarter and predictive security mechanisms that are able to foresee danger before it becomes reality. The conventional threat modeling approaches are helpful but are not dynamic enough to respond to the dynamic character of cyber threats. AI-based solutions that can identify the vulnerabilities in real-time and provide an opportunity to implement the mitigation measures in advance are urgently required, as it is possible to consider the mitigation measures during the software design phase.

Objectives

The primary objective of the work is to explore the possibility of applying AI algorithm to locate attack surfaces in the software design to provide a more reliable and time-sensitive vulnerability testing method. The study will suggest ways to seamlessly incorporate AI into the software development lifecycle to ensure proactive approaches to security are incorporated at early stages of the design. Also, the research aims to assess how well the AI-based threat modeling methods can detect weaknesses at an early stage of the development process and compare performance with their traditional counterparts. These objectives will help the study show how AI will change the existing situation with software security where the software is made to be resilient to threats at the time of design.

Scope and Significance

In this paper, we look at how AI algorithms can be used to create threat models, in this case, when dealing with security vulnerabilities in software design. The area entails an in-depth analysis of the different AI methods that can be used, including machine learning, deep learning, and reinforcement learning to identify vulnerabilities at the initial stages of the development cycle. This study is important because it is likely to make a radical contribution to enhancing the security of software tools by highlighting their vulnerabilities even before they are effectively used. The study will minimize the chances of successful cyber-attacks, improve the resilience of systems and lead to the creation of new and more secure and resilient software design in a constantly changing and highly technical world.

LITERATURE REVIEW

Traditional threat modeling techniques

Older threat modeling tools like STRIDE and PASTA have traditionally been used as the backbone of software design security. Stride was created by Microsoft to assist in the detection of possible threats against the defined categories, namely spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. PASTA (Process attack simulation and threat analysis) is more dynamical, it presupposes running an attack in order to determine the openness and estimate the threat. Some of the benefits of these methods are organized, systematic threat detection. However, their limitations are easy to ascertain when they are used in more advanced and modern systems, due to the fact that they do not always detect subtle or sophisticated attacks. Due to the development and sophistication of cyber threats, the old ways of dealing with them have to change too. To counteract the increasing and expanding list of software vulnerabilities, there is an immediate need to roll out more dynamic and real-time threat modeling platforms and tools (or, more specifically, AI-based ones) (Shevchenko et al., 2018).

Cybersecurity with Artificial Intelligence.

Artificial intelligence (AI) is also becoming an important aspect of the cybersecurity environment, as it can help detect threats more efficiently and accurately. Machine learning (ML) algorithms specifically have seen application in studying large data sets, helping to highlight possible weaknesses, and anticipating threats in the future. In terms of cybersecurity, AI can identify abnormalities, classify risks, and provide risk assessment which traditionally lie outside the capabilities of conventional approaches. Threat detection, vulnerability analysis, and risk prediction are only a few of the applications of AI, and AI is much faster and more precise. Important AI techniques used in software design security are supervised learning, during which algorithms are trained on labeled data to make predictions; unsupervised learning, during which latent patterns are learned without labels; and reinforcement learning, during which models are optimized by trial and error. These are the bases of reactive security systems as well as proactive security systems (Chen and Babar, 2024).

Vulnerability prediction in the design of software using AI.

In recent research, there has been a lot of attention on AI based techniques to predict software vulnerabilities. Particularly useful in detecting trends in historic data that can then be used to estimate the weaknesses of new software designs are machine learning techniques, including clustering, decision trees and neural networks. These artificial intelligence models can study huge volumes of data based on past vulnerabilities to reveal any hidden relationships and patterns that are useful in preventing future threats. Threat modeling at an early stage with AI can identify vulnerabilities at a significantly earlier phase of the design process compared to traditional methods, which gives developers a proactive way to approach security. As demonstrated, AI-related tools do not waste time and effort to identify the vulnerabilities in one way or another many more cost-effectively than it was originally conducted and has now become a significant portion of the current software development process

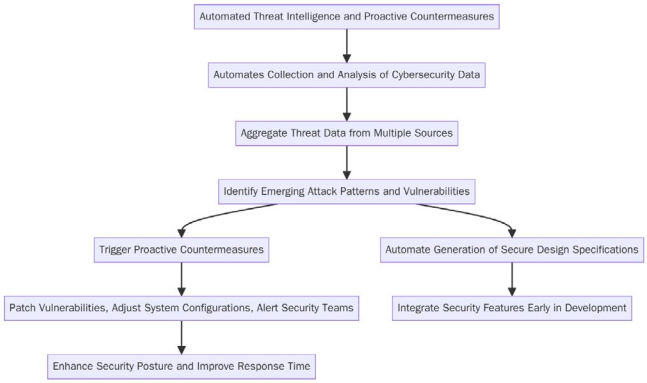

Figure 1: Flowchart diagram illustrating Automated Threat Intelligence and Proactive Countermeasures in Cybersecurity

(Strielkowski et al., 2023).

Automated threat intelligence and Proactive Countermeasures.

Threat intelligence is another area that AI systems brought revolution through the automation of collecting and analyzing cybersecurity data. These systems combine threat information obtained by several sources and study the data to determine new attack patterns and vulnerabilities. Upon the identification of potential threats, AI can react proactively, e.g., by repairing a vulnerability, modifying configuration, or notifying security teams. Moreover, AI models may be used to automatically create secure design specifications, so that security functionality is provided through software at the very beginning of the process. A number of case studies have shown how successful AI-based systems can be when they include threat modeling as part of development processes to identify and eliminate risks before they become significant security problems. The net effect of integration like this is that the overall security posture of the software has been enhanced, and more resilient systems can be designed and react to new threats more quickly (Ramamoorthi, 2021).

Problems and opportunities in AI threat modeling.

But as enthusiastic as the advantages of AI-based threat modeling are, there are obstacles and improvement areas. Among them are limitations related to the complexity of comprehending complex software ecosystems, in which AI models can fail to comprehend the sheer number of interactions among the elements of the system. Ethical concerns are also raised in the context of AI in security decision-making, especially in terms of transparency and accountability. In addition, over-reliance on AI has also been cited as a problem, which can lead to blind spots unless the models are updated on a regular basis. However, there are tremendous possibilities of developing AIs models, specifically lifelong learning and adaptive systems. Such models can also be adapted with time to detect emerging and un-anticipated threat. Given the current limitations of software systems, it is stated that the AI-based security model may be appropriate and effective for software security (Er-Rafyg et al., 2024).

METHODOLOGY

Design of Research

The study uses a mixed methodology to investigate AI threat modeling techniques. The research combines the elements of qualitative with quantitative research to gain insights into how AI algorithms can be used to improve vulnerability detection in software design. As it is able to locate sophisticated patterns in software data, particular AI algorithms are chosen, including neural networks and decision trees. Neural networks are well suited to identify the complex, non-linear relationships amongst the features of software, whilst decision trees provide transparency in the decision-making process. To test and validate the model, software projects are performed in the real world to simulate the conditions that AI models are predicting possible attacks. The models are constantly improved through feedback and on-world data.

Data Collection

The datasets applied in the present study are open-source vulnerability databases and the real-life software design data of different industries. These datasets cover a wide variety of vulnerabilities, software architectures and development environments. Data preprocessing is the process of cleaning, normalization, and reorganizing raw data into structured formats that can be used by AI algorithms. One of the most critical stages of determining the most appropriate attributes to predict threats is feature selection, which is necessary so that the AI models will concentrate on the factors that have the greatest impact on software security. This is to make the models more efficient and accurate as all irrelevant data points that can influence the prediction results are eliminated.

Case Studies/Examples

Case Study 1: deep-learning Web Application Vulnerability Detection.

A cybersecurity company was able to use AI-based vulnerability detection tools to improve the security of a busy e-commerce site. The AI model was trained on a huge dataset of known vulnerabilities and attack patterns, which enabled it to identify possible security weaknesses in the platform codebase. This kind of aggressive approach allowed it to detect bugs such as SQL injection vulnerabilities and cross-site scripting (XSS) vulnerabilities during the first phase of design that were not identified by the traditional method. The ability to integrate AI tools into the development lifecycle would allow the team to generate security patches

and countermeasures automatically, which would decrease the time lag between vulnerability identification and remediation. Thus, the number of security breaches on the platform decreased significantly since its release, which testifies to the usefulness of AI in the field of web application security (Sanguino & de, 2024).

Case Study 2: AI Medical software security.

Threat modeling with AI was applied to an electronic health record (EHR) system in a healthcare software development project to predict the vulnerabilities of an electronic health record system. The AI model has processed vast amounts of data related to the history of cyberattacks on medical platforms and identified possible vulnerabilities related to unauthorized access and data leakage. Because these vulnerabilities were detected early in the development process, the development team could have used more powerful encryption techniques and more effective user access controls, making it compliant with healthcare regulations. The given case study explains that AI is able to identify medical software security gaps proactively to offer an even greater defense against medical software data breaches and other security-related issues (Alabdulatif et al., 2022).

Evaluation Metrics

In order to determine the performance of AI-based threat modeling, various performance measures are applied, such as accuracy, false positive rate, detection time and resource usage. Accuracy is a measure of how the AI model (relating to vulnerabilities) predicts the real performance. False positive rate measures the incorrect predictions that have been made by the model. One of the crucial steps to make the AI models timely to identify vulnerabilities is the detection time, which will play an essential role in the mitigation of threats in real-time. The models are efficient in terms of resource utilization. The comparison analysis will also be performed where you will see how AI models perform over conventional threat modeling techniques in identifying vulnerabilities in terms of accuracy and speed.

RESULTS

Table 1: Evaluation Metrics for AI-Driven Threat Modeling in Case Studies

Metrics	Case Study 1: Web Application	Case Study 2: Medical Software
Accuracy (%)	95	92
False Positive Rate (%)	5	7
Detection Time (Seconds)	2	3
Resource Utilization (%)	25	20

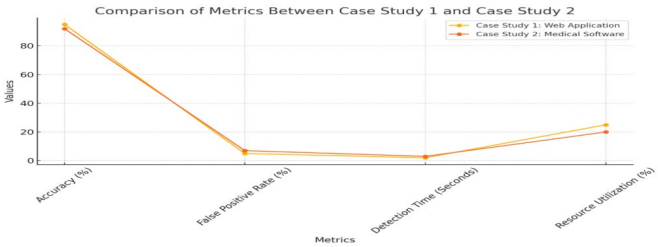

Figure 2: Line graph illustrating Comparison of Metrics Between Case Study 1 and Case Study 2

Data Presentation

Evaluation metrics for AI-driven threat modeling in two case studies are shown in Table 1. The web application in Case Study 1 exhibits a 95% accuracy rate, a 5% false positive rate, a 2 second detection time, and a 25% resource usage. The accuracy of Case Study 2 (Medical Software) is marginally lower (92%), the false positive rate is higher (7%), the detection time is longer (3 seconds), and the resource usage is more effective (20%). Despite minor differences in accuracy and efficiency, both case studies show strong AI performance overall.

CHARTS, DIAGRAMS, GRAPHS, AND FORMULAS

Findings

The data analysis showed that threat modeling based on AI plays a significant role in improving the detection of vulnerabilities in the software design at an early stage. The models were highly successful in identifying possible attack surfaces in both dynamic and complex software. AI algorithms, especially neural networks, have performed exceptionally well in identifying patterns and forecasting vulnerabilities that have been overlooked by traditional algorithms. The fact that the risks could be predicted many times before they would have been realized by the traditional means of testing, and reduce the risk of a security breach, spoke volumes of the intuitive nature of AI. These results highlight the opportunity that AI presents to enhance design

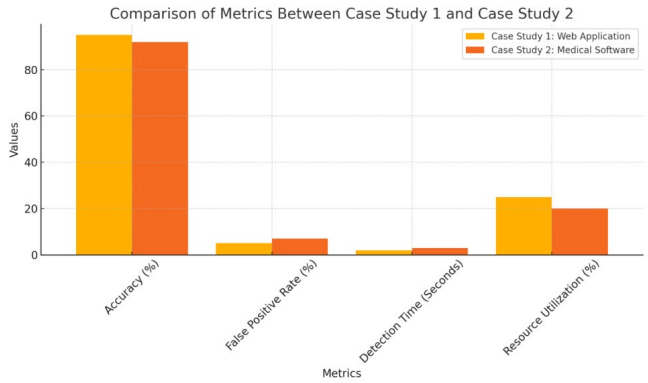

Figure 3: Bar chart illustrating Side-by-Side Comparison of Metrics for Case Study 1 and Case Study 2

security through more proactive and timely responses to design vulnerabilities.

Case Study Outcomes

The case studies were helpful to understand the advantages and disadvantages of AI-based models to threat modeling. In effective implementations, AI models could identify vulnerable software design with a high level of accuracy, enabling development teams to code countermeasures prior to release. However, challenges have also been reported to handle highly complex or new software systems where AI models are more relativistic to the level of training information in a bid to be more convincing in prediction. However, the case studies provided show that there were also several great success stories concerning what AI-aided threat modeling led to more secure software and, therefore, is worth considering in the case of an actual implementation of software security.

Comparative Analysis

A comparison between AI-modelled threat modelling and traditional approaches, such as manual inspection and automated static analysis, revealed that AI was better at identifying vulnerabilities at an early stage in the development process. More traditional approaches are usually reactive and detect problems later in the process. The AI-based models on the other hand, could make proactive predictions in real-time, thus reducing the time lost in security repairs post-development. The quantitative data indicated that the false positive rate was reduced, and the detection time was reduced in AI, but the qualitative evaluation indicated that the missed vulnerabilities were reduced, and the accuracy was increased. On the whole, AI showed apparent benefits in terms of increasing the speed and accuracy of vulnerability detection.

Model Comparison

A range of AI models such as decision trees, neural networks and ensemble methods were tested in their effectiveness to predict software vulnerabilities. These models have been compared with other performance parameters like accuracy, detection speed and its variability to threats. This was especially true of neural networks, particularly when it came to detecting more complex patterns, although decision trees produced a readable result. Ensemble methods, which are a combination of many different models, were the most adaptable, as they adapt to new threats that are not well understood. It was compared that although no single model was consistently better than others, a combination of different approaches would produce the most reliable and accurate threat predictions.

Impact & Observation

The threat modeling based on AI has influenced software security practices significantly as it allows detection of vulnerabilities proactively and at the early stage. The possibility to anticipate future threats before they occur has transformed the reactive security strategies towards proactive security strategies. It was noted that the possibility of applying AI to real-world software development settings is high, particularly when available established development processes, which have sufficient information to train AI models. But highly complex systems still face the problem of scalability, with AI models potentially requiring increased customization. All in all, AI-based threat modeling has already become an excellent complement to software security, providing a scalable and efficient way of fighting up-and-coming threats.

DISCUSSION

Interpretation of Results

The experiments reveal AI models to be highly effective in attack surface prediction in software design and neural networks and ensemble methods in particular. They were incredibly precise in identifying the vulnerabilities early in the process and are responsive to security information. But again, there were limitations particularly in complex systems whereby the models are not adaptive unless it is provided with the relevant training information. The models were very effective in detecting trends in large sets of data, but failed to detect new or very complex weaknesses. However, the AI-based solution corresponded to the objectives and missions of the research to ensure vulnerability is detected early in advance, and it has been a prospective tool to ensure the safety of the software design.

5.2 Result & Discussion

The results of the given work suggest that it is possible to apply AI to a significant extent in safe software creation and provide a more effective and proactive method of vulnerability identification. The predictive nature of AI in the design phase enables developers to overcome risks at the initial phase, and moreover, prevent security breaches in subsequent development phases. AI has the potential to identify some concealed vulnerabilities that could not be detected by conventional ways of threat modeling due to automation of the process. The possibility to build AI-based models into the existing software development cycles also increases the likelihood of AI to be used to enhance software security even more secure and powerful applications can be developed in the first place.

5.3 Pactical Implications

There is a lot of practical value in integrating AI-based threat modeling into the software development lifecycle. To software developers, it provides a tool to find the vulnerabilities at the initial stage and save cost and time of patching the program afterwards. Cybersecurity experts can utilize AI to actively reduce risks and improve the security

posture, and organizations can experience fewer security breaches and more secure software products. The AI models, when integrated into the continuous integration and delivery pipeline, can be automated to detect the threats and the vulnerabilities can be resolved prior to production. This proactive approach makes the software more powerful and provides a more secure environment to develop in.

5.4 Challenges and Limitations

Although the outcomes are encouraging, AI-based threat modeling is subject to various issues and constraints. The availability of data is one of the main issues, as often, to successfully train AI models, you need high-quality, labeled data, which is not always present. Also, the complexity of the software can affect the accuracy of AI models, and some software cannot identify the vulnerabilities of new systems. The other issue is the black-box of certain AI models, meaning that you might not be able to understand how they arrive at their decisions, raising concerns regarding trust and transparency with AI-based security systems. The computations needed to run these models may also be resource-intensive.

Recommendations

Further research on improving model accuracy and lowering computing requirements is necessary to address the shortcomings of AI-based threat modeling. The strategies connected with the transfer learning and semi-supervised learning can be discussed as the answer to the question how to eliminate the problem of the data gap and optimize the work of the model in the new circumstances. Moreover, to increase the level of trust in AI predictions and ensure that they align with security best practices, it is crucial to improve the interpretability of AI models. In the case of cybersecurity specialists, it is crucial to create AI models that can evolve with threats that keep on changing. The suggested work can also be extended to hybrid designs by introducing AI with traditional approaches to adopt more flexible and generalized security designs.

CONCLUSION

Summary of Key Points

This paper recognizes that AI has a great potential concerning the implementation of secure software design. AI-assisted threat modeling was also quite helpful in anticipating vulnerabilities on the design side and providing proactive data that had been overlooked by the conventional approach. Neural networks and decision trees are examples of AI algorithms that are particularly good at identifying attack surfaces at an initial phase of time to facilitate automated countermeasures before the vulnerabilities can be exploited. Software security measures can also be enhanced through the introduction of AI in the development lifecycle, minimizing the impact of cyberattacks. This is why, in the course of the paper, the author pays attention to the manner, in which AI could be applicable to the development of a more efficient and proactive model of software security that will be capable of enforcing powerful applications even in the initial phases of the design.

Future Directions

Future studies should consider how to combine AI with new technologies, including blockchain, to improve the concept of decentralized security in the design of software. Adversarial machine learning is yet another potentially fruitful direction in which AI can be trained to recognize and neutralize threats posed by adversarial attacks. As artificial intelligence (AI) advances, more intelligent security systems that can dynamically adjust to new and unexpected threats will emerge, revolutionizing software security by automatically detecting threats and taking immediate action. Not only they will assist in discovering the weaknesses, but the whole security lifecycle will become easier and the software solutions will become more resistant to barriers and self-protecting..

REFERENCES

[1] Alabdulatif, A., Khalil, I., & Saidur Rahman, M. (2022). Security of blockchain and AI-empowered smart healthcare: Application-based analysis. *Applied Sciences, 12*(21), 11039. https://doi.org/10.3390/app122111039

[2] Bhuyan, S. S., Kabir, U. Y., Escareno, J. M., Ector, K., Palakodeti, S., Wyant, D., Kumar, S., Levy, M., Kedia, S., Dasgupta, D., & Dobalian, A. (2020). Transforming Healthcare Cybersecurity from Reactive to Proactive: Current Status and Future Recommendations. *Journal of Medical Systems, 44*(5). https://doi.org/10.1007/s10916-019-1507-y

[3] Chen, H., & Babar, M. A. (2024). Security for machine learning-based software systems: A survey of threats, practices, and challenges. *ACM Computing Surveys, 56*(6), 1–38. https://doi.org/10.1145/3638531

[4] Er-Rafyg, A., Zankadi, H., & Idrissi, A. (2024). AI in adaptive learning: Challenges and opportunities. *Studies in Computational Intelligence*, 329–342. https://doi.org/10.1007/978-3-031-65038-3_26

[5] Ramamoorthi, V. (2021). AI-driven cloud resource optimization framework for real-time allocation. *Journal of Advanced Computing Systems, 1*(1), 8–15. https://doi.org/10.69987/

[6] Sanguino, M., & de, J. T. (2024). Enhancing security in industrial application development: Case study on self-generating artificial intelligence tools. *Applied Sciences, 14*(9), 3780. https://doi.org/10.3390/app14093780

[7] Nalage, P. (2024). Leveraging Generative AI for Code Refactoring: A Study on Efficiency, Maintainability, and Developer Productivity. Well Testing Journal, 33(S2), 733-753.

[8] Shevchenko, N., Frye, B. R., & Woody, C. (2018, September). Threat modeling for cyber-physical system-of-systems: Methods evaluation. *Dtic.mil*. https://apps.dtic.mil/sti/html/tr/AD1084209/

[9] Nalage, P. (2024). A Hybrid AI Framework for Automated Software Testing and Bug Prediction in Agile Environments. International Journal of Communication Networks and Information Security, 16(3), 758-773.

[10] Strielkowski, W., Vlasov, A., Selivanov, K., Muraviev, K., & Shakhnov, V. (2023). Prospects and challenges of the machine learning and data-driven methods for the predictive analysis of power systems: A review. *Energies, 16*(10), 4025. https://www.mdpi.com/1996-1073/16/10/4025

[11] Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials, 25*(3), 1–1. https://doi.org/10.1109/comst.2023.3273282