Sparking Light Publisher

Sparklinglight Transactions on Artificial Intelligence and Quantum Computing

journal homepage: https://sparklinglightpublisher.com/

ThreatVisionEval: Evaluating Multimodal Large Language Models for Threat Modeling Architecture Diagram Understanding

Santosh Pai[a,*], Srinivasa Rao Kunte R[b]

[a]*Research Scholar, Institute of Computer Science and Information Science, Srinivas University, Mangalore, India*
[b]*Research Professor, Institute of Computer Science and Information Science, Srinivas University, Mangalore, India*

Abstract

Threat Modeling is essential for identifying cybersecurity threats in software before implementation. It is a cornerstone of secure development. Currently, most threat modeling tasks are performed manually. Often human practioners have to recreate the threat modeling diagrams using archtiecture diagrams leading to delays. In practice, business teams often provide diagrams with ambiguous or unclear details. Advances in Artificial Intelligence (AI) have enabled multimodal Large Language Models (LLMs) to process both text and images. These models can extract security-relevant information from raw architectural diagram images. However, existing benchmarks for these models primarily assess general visual reasoning rather than security-specific capabilities. Key elements for threat modeling, such as entities, assets, call flows, trust boundaries, threat actors, and security properties, are missing from current LLM benchmarks.

This paper introduces ThreatVisionEval, a conceptual evaluation framework for multimodal LLMs. These are AI models capable of analyzing both images and text for vision-based threat modeling. The five core elements in the framework are : (1) a hierarchical task taxonomy,covers element and security property detection (2) a diagram variability model, handling notation, clarity, completeness, domain, and complexity; (3) a ground-truth annotation schema based on STRIDE elements (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege); (4) a metric suite including per-category F1 scores, Flow-Triple F1 for flow extraction, Boundary-IoU for boundary detection, and the Threat-Modeling-Readiness Score (TMRS); and (5) a prompt library with example questions and instructions, covering zero-shot, chain-of-thought, security-persona, and few-shot protocols.

By incorporating these five core elements, ThreatVisionEval offers clear definitions, reproducible protocols, and a practical roadmap. The framework enables systematic comparison of vision-language models for automated threat modeling. The paper also presents a research agenda. It recommends that the security, AI, and research communities adopt ThreatVisionEval as the standard for evaluating security-critical diagram understanding. This will facilitate accelerated progress and ensure robust, consistent outcomes.

Keywords: Threat modeling; Artificial Intelligence; Machine learning; Cybersecurity

1. Introduction

Threat modeling is recognized as an effective security practice by established frameworks such as OWASP (Open Web Application Security Project), NIST SSDF (National Institute of Standards and Technology Secure Software Development Framework), and Microsoft's Security Development Lifecycle [1][2][3]. However, fewer than 30% of development teams implement threat modeling systematically, and even fewer maintain updated models [4].

Email: g.santoshpai@gmail.com, kuntesrk@gmail.com

Please cite this article as: Santosh Pai et al., ThreatVisionEval: Evaluating Multimodal Large Language Models for Threat Modeling Architecture Diagram Understanding, Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (2025), 5(2), 89-98. **ISSN (Online):2583-0732.** Received Date: 2025/12/01, Reviewed Date: 2025/12/16, Published Date: 2024/12/24.

The challenge is not the lack of methodology; frameworks such as STRIDE (Spoofing, Tampering, Repudiation, Denial of Service, Elevation), LINDDUN (Linking, Identifying, Non-repudiation, Detecting, Data Disclosure, Unawareness, and Non-compliance), PASTA (Process of Attack Simulation and Threat Analysis), and Attack Trees provide comprehensive approaches. Primary challenge in automation of threat modeling is related to manual activities involved. The diagram creation in the threat modeling software is one of the time-consuming activities, as it involves re-creating or enhancing the existing architecture diagram to a format accepted by the threat modeling automation framework.

Multimodal LLMs can now process both images and text. This opens new avenues for automating threat modeling. Models such as GPT-4V, Claude-3.5-Sonnet, and Gemini-1.5-Pro can convert software diagrams into structured representations. These models work well with simple diagrams, but real-world diagrams from companies often include extra marks and mixed styles, making them hard to analyze.

This paper fills that gap by introducing a framework for evaluating multimodal LLMs in vision-based threat modeling which is named in our research as "ThreatVisionEval". Unlike general visual question answering, ThreatVisionEval focuses on extracting core elements needed for STRIDE threat modeling. The framework includes a task taxonomy, a diagram variability model, a JSON ground-truth schema, a comprehensive metric suite comprising the Threat Modeling Readiness Score (TMRS), and a prompt library for practical use. Reproducible guidelines enable model comparison and highlight deployment challenges.

2. Literature Review

From the previous research conducted on threat modeling automation we have identified primary challenge as substantial manual effort required to produce accurate diagrams, which often results in ambiguous or outdated representations [5][6] [7] [8] [9]. The advancements in AI and LLMs have greatly helped to automate various aspects of threat modeling.

We reviewed the existing research works in benchmarking the LLMs. Existing diagram-understanding benchmarks give limited insight for security practitioners. Datasets such as AI2D [10], ChartQA [11], FigureQA [12], and DocVQA [13] were made for general visual reasoning, document layout analysis, or software comprehension. These are not built for security applications. Such benchmarks primarily evaluate the extraction of functional components and their relationships. They omit critical security elements, including trust boundaries, authentication, and encryption on flows, explicit threat actors, and differences between protected assets and regular data stores. High performance on these benchmarks does not ensure that models can identify all STRIDE threat categories. For example, they may miss spoofing across trust boundaries or information disclosure over unencrypted channels. Because benchmark goals differ from security needs, the adoption of multimodal LLMs in security is inconsistent. Despite claims of "automated threat modeling from diagrams" in research and industry, there is no standard evaluation protocol to validate such claims.

[14] proposes a benchmarking method for LLMs based on privacy knowledge within the models, covering eight privacy aspects. [15] presents a benchmarking method focused on Swedish medical domains, emphasizing clinical inference relevance. Eighteen LLMs were evaluated using this benchmark; three ranked highest. [16] introduced sixty multiple-choice questions that assess areas covered by several benchmarks, enabling quick reviews for LLM adopters. The questions are designed to be completed by humans in one hour. [17] defines an accelerated benchmark for LLM inference by breaking it into multiple stages, with metrics for each. This benchmark is primarily used for chatbots and live translators. [18] describes a model that helps evaluate detection systems that differentiate between human- and LLM-written text. The benchmark helps develop systems that support ethical use of AI in education and science.

While these benchmarks cover various domains, none specifically target security threat modeling. This work will investigate the application of LLMs to security threat modeling.

3. ThreatVisionEval: The Proposed Evaluation Framework

Before detailing the technical components, it is essential to outline the guiding principles that inform the design of ThreatVisionEval and differentiate it from generic diagram-understanding benchmarks.

The framework is designed with security as its primary focus. Most existing datasets keep diagrams in their original form. In contrast, ThreatVisionEval extracts information essential for threat modeling. Standards such as Microsoft STRIDE, OWASP Threat Modeling Playbook [19], MITRE ATT&CK for Enterprise [20], and NIST SP 800-53 [21] are used to create a detailed list of elements needed for threat modeling. This ensures the framework checks a model's ability to detect these key elements, even if the input diagrams are incomplete, unusual, or unclear.

Second, ThreatVisionEval has prioritized reality over ideal datasets. Many benchmarks use refined images of diagrams, whereas in actual security assessments, this is rarely the case. The framework expects different types of diagrams, including whiteboard photos, low-quality images from presentations, outdated files with missing elements, and screenshots. To better align with the realities faced by Threat modelers, the framework considers visual noise, missing labels, and confusing notations as crucial evaluation criteria. This ensures we push models to demonstrate not just ideal performance on clean images, but also robustness. Adoptability and reproducibility are central design principles of the framework. Each component is thoroughly documented and provided in machine-readable formats, including task definitions, JSON schemas, metric formulae, and prompts. This level of detail enables a single researcher with moderate security expertise to efficiently construct a dataset of 60 diagrams, in contrast to the substantial effort required by frameworks such as AI2D or DocVQA [22] [23].

The framework is designed to be extensible, versioned, and future-proof, allowing for community-driven extensions without compromising the comparability of results from prior evaluations. The strength of the benchmark proposed lies in the five pillars present in the framework. These pillars work collectively to make the framework security focused benchmark. The Hierarchical task taxonomy pillar evaluates the component detection capabilities in the LLM with five levels that cover basic element detection to security property extraction. Diagram variability model evaluates LLMs ability to not only extract elements from ideal clean images, but a diverse dataset of real-world noisy architecture diagrams. The ground truth annotation schema ensures evaluators build the gold dataset in a standard consistent format that is ready for machine processing. Dedicated metric suite is an adoption of generic metrics for threat modeling focused evaluations of LLMs. The final pillar is a set of standard LLM prompts that ensure evaluations are fair and unbiased across different LLMs.

3.1. Hierarchical Task Taxonomy

Hierarchical Task Taxonomy in ThreatVisionEval is listed in Table 1. The multimodal LLM capabilities are organized into five progressive levels, L1 to L5. The levels represent the real-world threat modeling tasks, starting from basic component detection at L1 to security property extraction at L5. The five layers are designed to ensure the framework provides granular, actionable evaluations that mimic the human threat modeler's reasoning abilities.

Table 1. Hierarchical Task Taxonomy

Level Identifier	Task Description	Importance in Threat Modeling
L1	Detect and classify components (processes, data stores, external entities)	Foundational input to form the complete attack surface allowing threat enumeration
L2	Extract directed data/call flows with protocols	Essential for detecting information disclosure and repudiation threats
L3	Identify trust boundaries and zone membership	Crucial for finding spoofing, elevation of privilege, and lateral movement threats
L4	Infer threat actors (even when not explicitly marked as malicious)	Provides insights about external attacker entry points in the system
L5	Extract security properties (authentication mechanism, encryption, integrity checks)	Important for mitigation generation, and threat detection of Spoofing, Tampering, Information Disclosure categories

To pinpoint the exact failure mode, the evaluation must report performance separately at each level (L1-L5) for the LLMs.

3.2. Diagram Variability Model

The Diagram Variability Model is a distinctive pillar of the framework, ensuring evaluations reflect the complex, ambiguous conditions typical of human-involved threat modeling sessions. Unlike generic frameworks that rely on labeled, cleaned images for benchmarking [24] [25] [26], ThreatVisionEval incorporates six orthogonal sources of real-world degradation in architecture diagrams. This approach requires models to handle

balanced samples across Data Flow Diagrams (DFDs), C4, Unified Modeling Language (UML), and informal architecture sketches [27] [28] [29] [30].

Table 2 shows the different axes in the Diagram Variability Model. The Visual Quality property assesses sensitivity to compression, scanner noise, and overlapping handwriting. Completeness evaluates whether the model hallucinates or remains conservative when critical security elements are absent. The Domain property prevents overfitting to specific domains. The framework expects models to extract domain-specific information during evaluation. Complexity scaling identifies performance degradation in large diagrams. The input architecture may be highly complex, containing numerous elements typical of enterprise-scale diagrams. Legend and Annotation variation tests the model's ability to infer key security information when details are partially or entirely missing.

Table 2. Axis in Diagram Variability Model

Axis	Categories / Values	Purpose in Threat Modeling Evaluation
Visual Quality	Clean vector, Screenshot (compression), Scanned/hand drawn, Heavily overlapping/faded text	Measures robustness to real-world capture artefacts
Completeness	Complete, Missing components, Missing flows, Contradictory/misleading arrows	Evaluates hallucination control and conservative reasoning
Domain	Web applications, Cloud-native/microservices, IoT/OT/SCADA, Mobile backend, Payment systems	Prevents domain over-fitting; tests transferability
Complexity	Small (≤10 elements), Medium (11–25 elements), Large (>25 elements)	Exposes scalability limits in large enterprise architectures
Legend & Annotation	Legend fully inside image, Legend cropped/outside image, No legend at all	Tests symbolic reasoning and resistance to missing contextual keys

This multi-axis design ensures that models achieving high scores are genuinely effective for security threat modeling. A minimum of 60 architecture diagrams, stratified across the six categories [31], is required to evaluate model capabilities thoroughly.

3.3. Ground Truth Annotation Schema

The purpose of the Ground Truth Annotation Schema is for the evaluator to prepare a golden JSON structure for each input architecture image. The golden JSON is then compared with the JSON generated by the LLM. Comparing the golden JSON with LLM generated JSON will reveal how accurately the LLM has parsed the input architecture diagram image.

The schema specifies a minimal set of fields to represent architectural diagrams, including diagram_id, assets, entities, trust_boundaries, data_flows, and threat_actors, ensuring unambiguous labeling. Its compact design reduces the effort required to generate reference JSONs, making it feasible for evaluators with limited resources to build substantial datasets. Despite its simplicity, the schema remains sufficiently expressive to capture all necessary elements for automated threat enumeration.

Another advantage of such a standardized schema is that it avoids subjective human judgment. The standardized schema mitigates subjective human judgment and enables automated scoring during evaluation. Enforcing this schema ensures consistent scoring across various LLMs, enhancing the reliability of the proposed framework.

The schema is defined as below in JSON format:

```
{
  "diagram_id": "string",
  "source_description": "optional free text",
  "assets": [
    "string"
  ],
  "entities": [
    {
```

```
            "name": "string",
            "type": "process|datastore|external_entity",
            "trust_zone": "string"
         }
      ],
      "trust_boundaries": [
         {
            "name": "optional string",
            "zones": [
               "Internet",
               "DMZ",
               "Internal"
            ]
         }
      ],
      "data_flows": [
         {
            "source": "string",
            "target": "string",
            "protocol": "optional string",
            "authentication": "none|basic|mutual_tls|oauth2|jwt|kerberos|certificate",
            "encryption": "boolean",
            "integrity": "boolean optional"
         }
      ],
      "threat_actors": [
         "string"
      ]
   }
```

3.4. Dedicated Metric Suite

We have designed a quantitative measure for ThreatVisionEval to objectively assess how well the multimodal LLM understands input architecture diagrams, as required for threat modeling. Five metrics are defined in this suite, as shown in Table 3.

Table 3. Dedicated Metric Suite

Metric Identifier	Metric Name	Definition / Computation	Rationale (Why It Matters for Threat Modeling)
1	Asset / Entity / Threat-Actor F1	Token-level F1 with fuzzy matching (ratio ≥ 0.90)	Tolerates minor OCR/spelling variations
2	Flow-Triple F1	Exact (source, target, protocol); 0.5 credit if protocol missing	Protocol labels are frequently omitted
3	Trust-Boundary IoU	Jaccard index over zone pairs	Boundaries are set-based
4	Security-Property Accuracy	Exact match on authentication type and encryption/integrity flags	Directly determines the correctness of spoofing & disclosure analysis
5	Threat-Modeling-Readiness Score (TMRS)	(0.35 × Flow_F1 + 0.30 × Boundary_IoU + 0.20 × (Asset+Entity+Actor)_F1 + 0.15 × Property_Acc)	Single comparable headline metric

The first metric, 'Asset / Entity / Threat-Actor F1', evaluates the model's ability to extract element names from

architecture diagrams accurately. Tolerance is incorporated to account for common Optical Character Recognition (OCR) errors and spelling variations in diagram labels.

The second metric is 'Flor-Triple F1', which rewards models for correctly detecting flow direction and connectivity. A tolerance is included to forgive label omissions in the input diagrams.

The third metric, 'Trust Boundary IoU (Intersection over Union)', uses the Jaccard index [32] applied to zone pairs (e.g., "Internet" and "Public cloud"). This approach treats boundaries as sets rather than pixels, enhancing robustness to variations in dashed line styles. This metric is essential for detecting threats associated with lateral movement.

The fourth metric, "Security-Property Accuracy," answers simple yes-or-no questions about Encryption and Authentication between the elements. The metric considers these properties as the most critical from a security perspective. This metric checks whether AI can adequately understand the current encryption and authentication between elements, ensuring these vital properties are not misinterpreted.

The fifth metric, TMRS (Threat-Modeling-Readiness Score), aggregates the previously discussed scores into a single value ranging from 0 to 1. An LLM achieving a TMRS of 0.85 or higher demonstrates performance comparable to that of human experts in understanding architectural diagrams. The TMRS is calculated using formula 1.

$$TMRS = (0.35 \times Flow\_F1 + 0.30 \times Boundary\_IoU + 0.20 \times (average\ of\ Asset/Entity/Actor\ F1) + 0.15 \times Property\_Acc) \quad (1)$$

Overall, this pillar transforms subjective diagram parsing into objective, threat-focused benchmarking, paving the way for reliable AI-assisted threat modeling.

3.5. Standardized Prompt Library

One challenge in the evaluation is the lack of standardized prompts for inference with LLMs. If allowed to use tricky or tweaked prompts, the evaluation results will not be satisfactory. To ensure consistency, our proposed framework includes standardized prompts. These prompts ensure the evaluation is not tied to the prompt but to the LLM's ability to detect the security-related elements of the actual architecture diagram.

The library uses four prompts detailed in Table 4.

Table 4. List of Standardized Prompts

Standard Prompt	Description
Zero Shot Expert	No examples are provided to LLM in this prompt.
Chain-of-Thought	Explicitly tell the model to reason step by step for threat detection
Security-Persona	Places the model in the role of a Senior application security architect
Few-Shot	Provide three fully annotated input diagrams and output JSON examples, and then the actual input diagram.

These prompts encompass scenarios ranging from a complete cold start to those encountered by security experts in production settings. The prompts are standardized across models and research groups, ensuring fair and consistent evaluation. Detailed prompt descriptions are provided below.

- Zero Shot prompt template

```
You are an expert threat modeler. Looking only at the provided architecture diagram image, extract in strict JSON format using the following schema:

{
  "assets": ["..."],
```

```
  "entities": [{"name": "...", "type": "process|datastore|external_entity", "trust_zone": "..."}],
  "trust_boundaries": [{"name": "...", "zones": ["...", "..."]}],
  "data_flows": [{"source": "...", "target": "...", "protocol": "...", "authentication":
   "none|basic|jwt|oauth2|mutual_tls|...", "encryption": true|false}],
  "threat_actors": ["..."]
}

Include ONLY what is clearly visible in the diagram. Do not guess or add anything that is not shown.
```

- Chain-of-Thought prompt template

```
You are an expert threat modeler performing STRIDE analysis. Examine the diagram step by step:

1. List every box, circle, cloud, or labelled shape and classify it as process, datastore, or external entity.
2. For each shape, note the text label and any trust zone it sits inside (Internet, DMZ, Internal, etc.).
3. Trace every arrow/line, identify source and target, and note any protocol or lock icon.
4. Identify any dashed rectangles, shaded areas, or firewall icons that represent trust boundaries.
5. Note any external attacker icons or untrusted entities.

Now, based only on the above observations, output the extraction in the exact JSON schema provided in
the zero-shot prompt. Do not add anything that is not visible.
```

- Security-Persona prompt template

```
You are a senior application security architect with 15 years of experience performing STRIDE and
   LINDDUN threat modeling at Fortune-500 companies. Your job is to translate architectural diagrams
   into precise data flow diagrams for threat identification.

Looking at the provided diagram image, extract ONLY the elements that are clearly depicted using this
   exact JSON schema:

{same schema as above}

You are extremely conservative: if something is ambiguous or not shown, leave it out or mark it as
   unknown. Never hallucinate security properties.
```

- Few-Shot prompt template

```
You are an expert threat modeler. Here are three fully annotated examples of architecture diagrams and
their correct JSON outputs:

<Example 1 – image + correct JSON>
<Example 2 – image + correct JSON>
<Example 3 – image + correct JSON>

Now, using the same JSON schema and level of precision, analyze the following diagram and output only
the JSON.
<target diagram image>
```

4. Guidelines for Framework Instantiation and Dataset Creation

This section presents a practical, step-by-step guide enabling researchers and practitioners to produce a ThreatVisionEval-compliant dataset and evaluation within four weeks using minimal resources. Evaluators can utilize these guidelines to generate data, execute model evaluations, and publish stratification tables and TMRS values. Comprehensive instructions ensure comparability across studies.

4.1. Guideline on recommended dataset size and structure

Precise dataset size requirements have been established to ensure reliability and practical relevance. The minimum viable dataset comprises 60 architecture diagrams, while 100–150 diagrams are recommended for robust formal publication. Strict stratification is required to achieve comprehensive coverage across all variability axes.

A sample dataset of 100 diagrams should be distributed across domains and visual quality. For instance, 25 diagrams may originate from web applications, 25 from cloud-native applications, 25 from Internet of Things systems, 15 from the payment industry, and 10 from other sectors. Visual quality should also be varied, including 30 clean diagrams with precise data flows and labels, 30 screenshots from books and existing drawings, 20 handwritten or scanned diagrams, and 20 diagrams characterized by significant noise and overlapping elements.

This ensures the evaluation results are close to real-world assessments.

4.2. Guideline on diagram collection strategy

It is crucial to perform the evaluation, ensuring the process is legally compliant and ethical. The first and most important requirement is that the dataset must come from openly accessible sources explicitly licensed for research purposes, or the evaluator must have licenses from the diagram owners. One of the resources evaluators can use is public GitHub repositories, which typically include README files and architecture markdown files. Official documentation of widely used projects such as Kubernetes, Apache, and Istio can be readily used for the evaluation. Publicly available reference architectures from AWS (Amazon Web Services), NIST, or OWASP can be used for evaluation when specific architecture diagrams are required. In contrast, it is not recommended to use internal, confidential diagrams protected by organizational or customer licenses, or to use customer reports, as they cannot be published, and results may not be transparent to the users of the evaluation. However, for an organization's internal tool validation or evaluation, it is allowed to use such confidential diagrams as required.

4.3. Guidelines on annotation workflow

The annotation process is critical for ensuring high-quality, consistent ground-truth labeling in ThreatVisionEval evaluations. It is recommended to engage two independent annotators, at least one of whom has practical threat modeling experience, to create JSON files in accordance with the prescribed schema. Open-source tools such as Label Studio can expedite annotation, enabling completion in under four minutes per diagram. To assess inter-annotator agreement on entities and data flows, Cohen’s kappa [33] is employed; a score of 0.85 or higher indicates strong consensus, ensuring that gold-standard JSONs are reliable and not subject to individual bias.

4.4. Guidelines on the evaluation script and reproducibility package

For published evaluations, it is essential to provide a reproducibility package that enables independent verification and replication of results. The recommended package should include the diagram image dataset, gold-standard JSON annotation files, evaluation scripts, model-generated JSON responses, and model configurations (such as temperature settings). This package should be made publicly available alongside the results. A concise documentation file explaining the usage of each component will facilitate straightforward reproduction of evaluation outcomes.

4.5. Guidelines for selection of baseline models

To ensure fair model comparison, each ThreatVisionEval evaluation should include a standardized set of widely available baseline models, such as GPT-4o, Claude 3.5 Sonnet, Gemini 1.5 Pro, and Llama 3.2 Vision 90B. Evaluation planning should prioritize the use of the latest available models. Incorporating both established and new models enables consumers to assess advancements in image understanding capabilities relevant to threat modeling. Additionally, the cost of model usage should be considered during evaluation planning.

4.6. Guidelines on publication checklist

This section outlines best practices for preparing evaluations for publication. Evaluators should explicitly state that the ThreatVisionEval framework was employed, enabling consumers to understand the evaluation methodology. The full stratification table, demonstrating coverage across the six variability axes, must be included.

Complete TMRS score values, along with details of prompting techniques (e.g., zero-shot or chain-of-thought), should be reported. The dataset images, gold-standard JSON annotations, evaluation scripts, and raw model responses must accompany the evaluation to facilitate reproducibility and enhance transparency within the research community.

5. Conclusion

This paper introduced ThreatVisionEval, an LLM evaluation framework designed to assess the multimodal LLM's capabilities for understanding architectural diagrams. The framework closes critical gaps in existing general diagram-understanding frameworks by defining security-specific primitives. This shifts the evaluation paradigm towards real-world threat modeling needs. The framework proposed in the paper is designed using five pillars that includes a task taxonomy, diagram variability model, ground-truth schema, dedicated metric suite, and a prompt template library. These ensure a complete, reproducible, and extensible basis for assessing LLM's performance. The evaluation closely mirrors practical Threat modeling scenarios.

Along with the conceptual framework, the paper provides clear guidelines for practical evaluations. This helps researchers and practitioners realize evaluations using the ThreatVisionEval framework. The steps, including recommended dataset size and stratification requirements, ensure a critical assessment of the model's capabilities. The guidelines also address ethical and legal considerations for dataset creation. This ensures evaluators can use the framework safely. The baseline model list provided in the guideline serves as a starting point for triggering evaluation. Evaluators are encouraged to use the latest, more capable models when available. Overall, the guidelines are designed to remove barriers to adoption in the research community.

We are confident that this framework will become the benchmark for evaluating any LLM purporting to understand security-related architectural diagrams.

Our future work will apply the framework to multimodal LLMs and deliver a publicly available benchmark for threat modeling. This will enable the research community and organizations to confidently choose the best LLM for automating threat modeling.

References

[1] Soares Cruzes, D., Gilje Jaatun, M., Bernsmed, K., & Tondel, I. A. (2018). Challenges and Experiences with Applying Microsoft Threat Modeling in Agile Development Projects. 2018 25th Australasian Software Engineering Conference (ASWEC), 111–120. https://doi.org/10.1109/ASWEC.2018.00023

[2] Pai, S., & Kunte R., S. R. (2022). A Comprehensive Analysis of Automated Threat Modeling Solution Company: Threat Modeler Software, Inc. International Journal of Case Studies in Business, IT, and Education, 249–258. https://doi.org/10.47992/IJCSBE.2581.6942.0193

[3] Shi, Z., Graffi, K., Starobinski, D., & Matyunin, N. (2022). Threat Modeling Tools: A Taxonomy. IEEE Security & Privacy, 20(4), 29–39. https://doi.org/10.1109/MSEC.2021.3125229

[4] State of Threat Modeling. (2023). https://www.securitycompass.com/reports/the-2023-state-of-threat-modeling/

[5] Kim, K. H., Kim, K., & Kim, H. K. (2022). STRIDE‐based threat modeling and DREAD evaluation for the distributed control system in the oil refinery. ETRI Journal, 44(6), 991‐1003. https://doi.org/10.4218/etrij.2021-0181

[6] De Rosa, F., Maunero, N., Prinetto, P., Talentino, F., & Trussoni, M. (2022). ThreMA: Ontology-Based Automated Threat Modeling for ICT Infrastructures. IEEE Access, 10, 116514–116526. https://doi.org/10.1109/ACCESS.2022.3219063

[7] Chlup, S., Christl, K., Schmittner, C., Shaaban, A. M., Schauer, S., & Latzenhofer, M. (2022). THREATGET: Towards Automated Attack Tree Analysis for Automotive Cybersecurity. Information, 14(1), 14. https://doi.org/10.3390/info14010014

[8] Välja, M., Heiding, F., Franke, U., & Lagerström, R. (2020). Automating threat modeling using an ontology framework: Validated with data from critical infrastructures. Cybersecurity, 3(1), 19. https://doi.org/10.1186/s42400-020-00060-8

[9] Granata, D., Rak, M., & Mallouli, W. (2023). Automated Generation of 5G Fine-Grained Threat Models: A Systematic Approach. IEEE Access, 11, 129788–129804. https://doi.org/10.1109/ACCESS.2023.3333209

[10] AI2D-Architecture: A Benchmark for Evaluating Multimodal Large Language Models in Architecture. (2025). https://datahub.hku.hk/articles/conference_contribution/15_AI2D-Architecture_a_Benchmark_for_Evaluating_Multimodal_Large_Language_Models_in_Architecture/29349782/1/files/55631954.pdf

[11] Masry, A., Long, D., Tan, J. Q., Joty, S., & Hoque, E. (2022). ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. Findings of the Association for Computational Linguistics: ACL 2022, 2263–2279. https://doi.org/10.18653/v1/2022.findings-acl.177

[12] Kahou, S. E., Michalski, V., Atkinson, A., Kadar, A., Trischler, A., & Bengio, Y. (2017). FigureQA: An Annotated Figure Dataset for Visual Reasoning (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1710.07300

[13] Mathew, M., Karatzas, D., & Jawahar, C. V. (2020). DocVQA: A Dataset for VQA on Document Images (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2007.00398

[14] Shahriar, S., & Dara, R. (2025). Priv-IQ: A Benchmark and Comparative Evaluation of Large Multimodal Models on Privacy Competencies. AI, 6(2), 29. https://doi.org/10.3390/ai6020029

[15] Moëll, B., Farestam, F., & Beskow, J. (2025). Swedish Medical LLM Benchmark: Development and evaluation of a framework for assessing large language models in the Swedish medical domain. Frontiers in Artificial Intelligence, 8, 1557920. https://doi.org/10.3389/frai.2025.1557920
[16] Gignac, G. E., & Ilić, D. (2025). Psychometrically derived 60-question benchmarks: Substantial efficiencies and the possibility of human-AI comparisons. Intelligence, 110, 101922. https://doi.org/10.1016/j.intell.2025.101922
[17] Jurkschat, L., Gattogi, P., Vahdati, S., & Lehmann, J. (2025). BALI—A Benchmark for Accelerated Language Model Inference. IEEE Access, 13, 98976–98989. https://doi.org/10.1109/ACCESS.2025.3576898
[18] Le, L., & Tran, D. (2025). A Metric-Based Detection System for Large Language Model Texts. ACM Transactions on Management Information Systems, 16(1), 1–19. https://doi.org/10.1145/3704739
[19] Sebastien Deleersnyder. (2025, December 13). OWASP Threat Modeling Playbook (OTMP) [Online post]. https://owasp.org/www-project-threat-modeling-playbook/
[20] Xiong, W., Legrand, E., Åberg, O., & Lagerström, R. (2022). Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. Software and Systems Modeling, 21(1), 157–177. https://doi.org/10.1007/s10270-021-00898-7
[21] El Marzak, Y., Mansouri, K., & Faris, S. (2025). A Comprehensive Metamodel for Cybersecurity: Based on NIST SP 800-53 Revision 5 Security and Privacy Controls. In I. Aboudrar, F. Ilahi Bakhsh, A. Nayyar, & I. Ouachtouk (Eds.), Innovative Technologies on Electrical Power Systems for Smart Cities Infrastructure (pp. 268–280). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-86705-7_25
[22] Parthasarathy, R., Collins, J., & Stephenson, C. (2025). What Makes a Good Generated Image? Investigating Human and Multimodal LLM Image Preference Alignment (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2509.12750
[23] Hiippala, T., Alikhani, M., Haverinen, J., Kalliokoski, T., Logacheva, E., Orekhova, S., Tuomainen, A., Stone, M., & Bateman, J. A. (2021). AI2D-RST: A multimodal corpus of 1000 primary school science diagrams. Language Resources and Evaluation, 55(3), 661–688. https://doi.org/10.1007/s10579-020-09517-1
[24] Cheng, K., Song, W., Fan, J., Ma, Z., Sun, Q., Xu, F., Yan, C., Chen, N., Zhang, J., & Chen, J. (2025). CapArena: Benchmarking and Analyzing Detailed Image Captioning in the LLM Era (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2503.12329
[25] Li, B., Li, X., Lu, Y., & Chen, Z. (2024). LossAgent: Towards Any Optimization Objectives for Image Processing with LLM Agents (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2412.04090
[26] Chao, W.-L., Cheng, K., Chowdhury, A., Kil, J., Lee, J., Liu, Y., Mai, Z., Wang, L., & Wang, Z. (2024). MLLM-CompBench: A Comparative Reasoning Benchmark for Multimodal LLMs. Advances in Neural Information Processing Systems 37, 28798–28827. https://doi.org/10.52202/079017-0906
[27] Mbaka, W. B., Zhang, X., Wang, Y., Li, T., Massacci, F., & Tuma, K. (2025). Assessing the usefulness of Data Flow Diagrams for validating security threats. Computers & Security, 156, 104498. https://doi.org/10.1016/j.cose.2025.104498
[28] Mavrogiorgou, A., Kiourtis, A., Kyriazis, D., Serrano, M., Isaja, M., Lazcano, R., Soldatos, J., & Troiano, E. (2025). C4 Model: A Research Guide for Designing Software Architectures. 2025 8th International Conference on Software and System Engineering (ICoSSE), 1–9. https://doi.org/10.1109/ICoSSE65712.2025.00009
[29] Carducci, M. (2025). Documenting Architecture. In M. Carducci, Mastering Software Architecture (pp. 361–398). Apress. https://doi.org/10.1007/979-8-8688-0410-6_24
[30] Tagliaferro, A., Corbo, S., & Guindani, B. (2025). Leveraging LLMs to Automate Software Architecture Design from Informal Specifications. 2025 IEEE 22nd International Conference on Software Architecture Companion (ICSA-C), 291–299. https://doi.org/10.1109/ICSA-C65153.2025.00049
[31] ElZemity, A., Arief, B., & Li, S. (2025). Analysing Safety Risks in LLMs Fine-Tuned with Pseudo-Malicious Cyber Security Data (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2505.09974
[32] Krasnodębska, K., Goch, W., Uhl, J. H., Verstegen, J. A., & Pesaresi, M. (2025). Advancing Precision, Recall, F-score, and Jaccard index: An approach for continuous, ratio-scale measurements. Environmental Modelling & Software, 193, 106614. https://doi.org/10.1016/j.envsoft.2025.106614
[33] Martín Andrés, A., & Álvarez Hernández, M. (2025). Estimators of various kappa coefficients based on the unbiased estimator of the expected index of agreements. Advances in Data Analysis and Classification, 19(1), 177–207. https://doi.org/10.1007/s11634-024-00581-x

******