

ThreatQwen: Domain-Adaptive Fine-Tuning of a 1.7B LLM for Security Event Classification and SOC Triage

Md. Minaruzzaman Shovon

`minar.svn@gmail.com`

Research Article

Keywords:

Posted Date: September 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9862964/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

ThreatQwen: Domain-Adaptive Fine-Tuning of a 1.7B LLM for Security Event Classification and SOC Triage

Md. Minaruzzaman Shovon
Chittagong University of Engineering & Technology
minar.svn@gmail.com

Abstract

Security operations teams sift through millions of detection-rule alerts every day, and the reasoning step that turns a raw alert into a triaged decision remains largely manual. Large frontier language models handle this kind of reasoning well (Achiam et al., 2023; Anthropic, 2024). Still, the organizations that need it most — banks, hospitals, defense contractors, and intelligence agencies — typically cannot send their logs to an external API. We present **ThreatQwen-1.7B-Detect**, a small open-source language model fine-tuned with QLoRA (Dettmers et al., 2023) on a curated multi-source corpus combining real detection rules from SigmaHQ (SigmaHQ, 2023), Elastic Detection Rules (Elastic, 2024), and Nuclei templates (ProjectDiscovery, 2024), together with GPT-4o-generated synthetic benign baselines covering Windows activity, web traffic, email, database, and VPN/remote-access events. On a fully balanced held-out test set of 882 examples (441 malicious, 441 benign), the model reaches **95.1% overall accuracy** with 96.1% malicious recall and 94.2% benign recall, substantially outperforming GPT-4o (51.8%) and GPT-4o-mini (51.9%) on the same examples, and improving its own untuned base (Yang et al., 2025) by 13 percentage points. The model runs locally on a single consumer GPU in approximately 3.1 GB of VRAM with no network dependency, making it directly deployable in air-gapped SOC environments where frontier APIs are unavailable. We release the model, dataset, and full training pipeline.

1. Introduction

A modern enterprise security operations center (SOC) ingests millions of telemetry events per day across endpoint, cloud, and web surfaces (Vielberth et al., 2020). Open detection-rule projects such as SigmaHQ (SigmaHQ, 2023), Elastic Detection Rules (Elastic, 2024), and Nuclei (ProjectDiscovery, 2024) trigger on suspicious patterns within that stream. Their output, however, is binary: an event either matches a rule or it does not. The reasoning step — deciding whether a triggered event represents a genuine threat, a known false positive, or something novel deserving closer attention — still falls entirely on the analyst. With alert volumes far outpacing analyst capacity (Alahmadi et al., 2022), that step is precisely where SOC workflows break down.

Frontier large language models can perform this kind of reasoning. GPT-4o (Achiam et al., 2023), Claude (Anthropic, 2024), and Gemini (Gemini Team, Google, 2023) all produce fluent, structured analyses of security events when prompted appropriately. The difficulty is that the environments that most need this assistance cannot use it. Regulated SOC handling HIPAA, PCI-DSS, FedRAMP, or defense data are typically prohibited from transmitting raw logs to external APIs (Tabassi, 2023). What they need is a small, locally deployable model that produces the same quality of structured analyst reasoning on their own hardware.

Existing security-focused LLM work has focused on general cybersecurity question answering (ElZemity et al., 2025; Iyer et al., 2025), pre-trained transformers for intrusion detection (Ferrag et al., 2023; Hassanin et al., 2024), or fine-tuning generalist assistants (Cisco, 2024; Otoum et al., 2025). To our knowledge, no prior work has fine-tuned a small open model on a unified, multi-source corpus drawn from several open detection-rule projects, augmented it with realistic GPT-4o-generated benign data, and evaluated it head-to-head against frontier APIs on structured event triage.

We make three contributions. **First**, we release ThreatQwen-1.7B-Detect, a QLoRA (Detrmers et al., 2023) fine-tune of Qwen3-1.7B-Instruct (Yang et al., 2025) that takes a raw security event and returns a structured JSON verdict. **Second**, we describe a multi-source training corpus that combines real detection rules with high-quality synthetic benign baselines generated using GPT-4o (Achiam et al., 2023), along with a reproducible data preparation pipeline that includes deduplication, severity normalization, and balanced sampling. **Third**, we evaluate the model head-to-head against GPT-4o and GPT-4o-mini on a fully balanced 882-example test set and show that it outperforms both frontier APIs by a wide margin while running locally on a single consumer GPU. The model, training pipeline, and dataset are publicly released to support reproducibility.

2. Dataset

2.1 Sources and Composition

The training corpus combines three real-world detection-rule projects with synthetically generated benign and ambiguous examples. Real-world entries are derived directly from open rule sets — SigmaHQ (SigmaHQ, 2023), Elastic Detection Rules (Elastic, 2024), and Nuclei templates (ProjectDiscovery, 2024) — while benign examples are generated programmatically using GPT-4o (Achiam et al., 2023) to ensure quality, diversity, and schema consistency.

The original dataset contains 8,454 raw examples. After deduplication by event text, normalization of null severity fields, and removal of examples with invalid or missing verdict labels, the cleaned corpus contains 5,610 unique examples. The verdict distribution is heavily skewed: approximately 71.6% of examples are malicious, and 28.4% are benign, reflecting the fact that open detection-rule projects describe attacker behavior almost exclusively (Elastic, 2024; SigmaHQ, 2023).

To address this imbalance, we augmented the benign class using GPT-4o (Azure) to generate 1,500 additional synthetic benign examples covering five distinct operational categories not well represented in the original data: Windows system and process activity, web and HTTP traffic, email and Exchange server events, database operations (SQL, PostgreSQL, MongoDB, Redis), and VPN and remote-access sessions (Cisco AnyConnect, OpenVPN, RDP, SSH, Okta, Azure AD). Each generated example follows the same JSONL schema as the original dataset and was verified to contain all required fields before inclusion. Synthetic augmentation of imbalanced security datasets has been shown to materially improve downstream classifier robustness (Chawla et al., 2002; Goodfellow et al., 2014).

2.2 Cleaning and Normalization Pipeline

Raw data passes through a five-stage cleaning pipeline before training. **First**, all examples are deduplicated using an exact event-text match, removing 2,844 duplicate entries (33.6% of the raw corpus) — a step consistent with best practices for LLM training corpora (Lee et al., 2022). **Second**, verdict labels are normalized to lowercase and filtered to the two valid labels — *malicious* and *benign* — discarding any example with an unrecognized or missing verdict. **Third**, null or unknown severity values are coerced to *none*. **Fourth**, the MITRE ATT&CK (MITRE

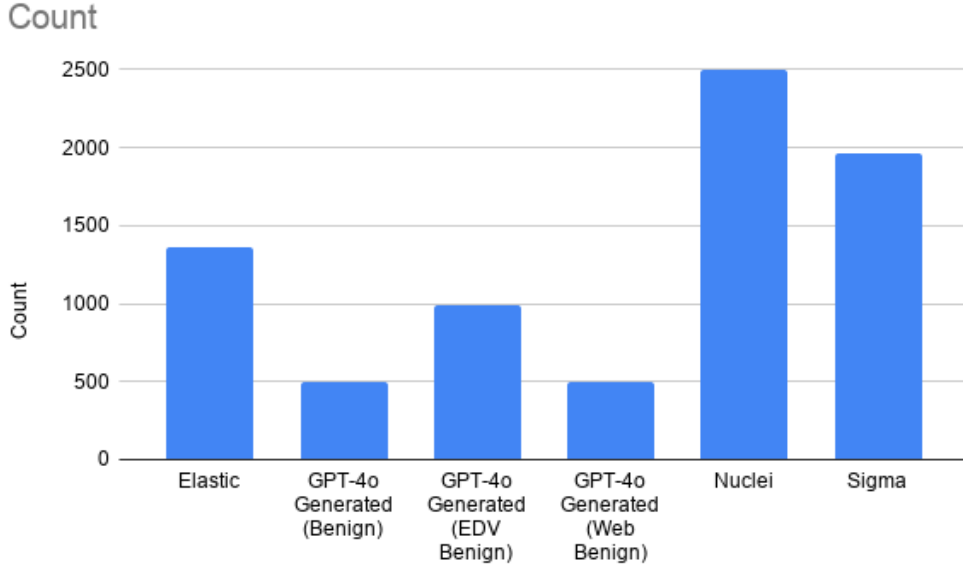


Figure 1. Composition of the multi-source training corpus after cleaning and augmentation.

Corporation, 2024) field is dropped entirely: 63.9% of examples carry empty MITRE arrays, the tags are inconsistently cased (T1059 versus t1059.001), and the field provides negligible training signal because the model output schema does not include it. **Fifth**, a sequence-length analysis confirms that p95 of token length across the corpus is 244 tokens and the absolute maximum is 821 tokens, so `MAX_SEQ_LEN` is set to 512 — sufficient to cover the entire corpus while halving attention compute compared to 1024 (Vaswani et al., 2017).

2.3 Synthetic Benign Generation

The original dataset contains 2,400 benign examples, all generated synthetically from templated rules. While effective for the first pass, this limited variety left the model struggling to generalize to benign events that superficially resemble malicious patterns. We designed a structured GPT-4o-based generation pipeline (Achiam et al., 2023) to produce higher-quality, more diverse benign examples, following the data-augmentation paradigm widely adopted in modern NLP (Feng et al., 2021; Wei & Zou, 2019).

Each generation call supplies GPT-4o with a specific operational scenario (for example, “Normal Cisco AnyConnect VPN login from known employee home IP” or “Normal PostgreSQL VACUUM ANALYZE running on large table”) together with a strict system prompt enforcing the JSONL output schema. Temperature is set to 0.95 to encourage lexical diversity. The model is instructed to include realistic log details: timestamps, source and destination IPs, usernames, hostnames, port numbers, process paths, and HTTP status codes where applicable. All generated examples have their `verdict`, `severity`, `miter`, and `level` fields programmatically set to *benign*, *none*, `[]`, and *none*, respectively, to prevent any label noise caused by the generative model.

Generation is parallelized across ten threads using Python’s `ThreadPoolExecutor`, reducing wall time by roughly 8× compared to sequential generation. Each batch targets a distinct scenario category, and scenarios are shuffled before assignment to prevent consecutive repetition. A JSON parse validation step discards any response that does not conform to the schema. Three separate generation batches were produced: 500 general Windows and system events, 500 web and HTTP traffic events, and 500 email, database, and VPN/remote-access events. Together, these add 1,500 high-quality, synthetic, benign examples to the corpus.

2.4 Train/Test Split and Class Balancing

To prevent source-level leakage between train and test (Kapoor & Narayanan, 2023), examples are partitioned using a stratified-source scheme: examples are grouped first by verdict label, then by source rule identifier, and source identifiers — not individual examples — are split 90/10 between training and test. No test example originates from a source rule seen during training. A post-split sanity check confirmed zero event-level overlap between the train and test sets.

The training set is class-balanced by oversampling benign examples 4 $\times$, yielding a training distribution with approximately equal counts of malicious and benign examples (He & Garcia, 2009). A 10% random subset of the balanced training set is held out as a validation split for monitoring training loss. The test set retains its natural source distribution and is not rebalanced during training; however, evaluation is conducted on a balanced 50/50 draw (441 malicious, 441 benign) to produce unbiased per-class metrics.

2.5 A Concrete Example

To make the pipeline concrete, consider a SigmaHQ rule (SigmaHQ, 2023) that flags encoded PowerShell launched from a Microsoft Office process. The parser extracts field constraints (`ParentImage` ends with `winword.exe`, `Image` ends with `powershell.exe`, `CommandLine` contains `-enc`), the associated MITRE tag (MITRE Corporation, 2024), and the severity level. The event synthesizer produces a Sysmon-style process-creation log satisfying those constraints:

```
EventID: 1 (Process Creation)
Image:   C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ParentImage: C:\Program Files\Microsoft Office\root\Office16\winword.exe
CommandLine: powershell.exe -nop -w hidden -enc SQBFAFgAIAAo...
User:    jsmith
Hostname: FIN-LAPTOP-12
```

The final training record is a single JSONL object that contains the event text, a reasoning trace explaining the detection logic grounded in the rule metadata, the verdict label, and the severity level. For the updated model, MITRE tags are omitted from the output schema to avoid training on 63.9% of examples in which this field is empty.

3. Method

3.1 Base Model

We fine-tune Qwen3-1.7B-Instruct (Yang et al., 2025) as the base. The Qwen3 family offers strong instruction-following at small parameter counts and handles syntactically dense inputs well — a practical requirement for security telemetry, which consists largely of command-line strings, registry paths, JSON CloudTrail records, and HTTP payloads. The model is loaded in 4-bit NF4 quantization via BitsAndBytes (Dettmers et al., 2022), reducing the memory footprint to approximately 3.1 GB at inference.

3.2 QLoRA Fine-Tuning

We use QLoRA (Dettmers et al., 2023) to attach trainable low-rank adapter matrices (Hu et al., 2021) to all major linear projections in both the attention and feedforward layers: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. Targeting all layers rather than attention-only is recommended by the original QLoRA paper (Dettmers et al., 2023) and consistently outperforms partial-layer approaches. LoRA rank is set to 16 with alpha 32 (a 2 $\times$ ratio, the recommended heuristic for strong update scaling) (Hu et al., 2021), dropout 0, and no bias terms. Training is accelerated using the Unsloth framework (Unsloth AI, 2024), which

provides optimized kernels for LoRA and QLoRA fine-tuning. The full set of hyperparameters is shown in Table 1.

Table 1. Training hyperparameters.

Hyperparameter	Value
Base model	unsloth/Qwen3-1.7B-unsloth-bnb-4bit
Quantization	4-bit NF4 (QLoRA)
LoRA Rank / Alpha	16 / 32
Target modules	q/k/v/o_proj, gate/up/down_proj
Epochs	3–5 (no early stopping)
Learning rate	2e-4
Effective batch size	16 (batch 4, accumulation 4)
Hardware	Tesla T4 (Kaggle free tier)

3.3 Output Schema

The model is trained to produce a compact JSON object containing exactly one field: **verdict** (*malicious* or *benign*). Both the severity and MITRE technique fields present in the original model card were removed. Severity acts as a near-perfect proxy for verdict in this dataset — every benign example has severity *none*, while every malicious example has a non-*none* severity — making it a data-leakage shortcut (Kaufman et al., 2012) rather than a genuine signal. MITRE tags (MITRE Corporation, 2024) were dropped because 63.9% of training examples contain empty arrays, which provide negligible training signal. The simplified single-field schema also reduces output token count and improves inference speed.

The system prompt instructs the model to respond only with valid JSON, with no additional text. During inference, leading newline tokens — a known artifact of Qwen3’s chat template (Yang et al., 2025) — are stripped, and the last non-empty block of the output is extracted before JSON parsing. A regex fallback extracts the verdict string if JSON parsing fails.

3.4 Class Imbalance Mitigation

The base dataset is 71.6% malicious. Without intervention, the model learns to predict *malicious* for nearly every input (Buda et al., 2018; He & Garcia, 2009). We apply a two-pronged strategy: 4× benign oversampling in the training set and removing the severity field from the output schema. The severity field is a near-perfect proxy for verdict in this dataset, so including it in the output creates a shortcut that the model exploits rather than learning to reason from event content (Geirhos et al., 2020). Removing it forces the model to attend to the event text itself.

4. Evaluation

4.1 Test Set and Baselines

Evaluation is conducted on a balanced held-out test set of 882 examples drawn equally from the malicious (441) and benign (441) pools of the stratified-source test split. The same 882 examples are presented to all four models: ThreatQwen-1.7B-Detect, the untuned Qwen3-1.7B-Instruct base (Yang et al., 2025), GPT-4o (Azure, zero-shot) (Achiam et al., 2023), and GPT-4o-mini (Azure, zero-shot).

GPU-based models (base and fine-tuned) are evaluated sequentially due to the single-GPU constraint. API models are evaluated in parallel across all test samples using a 10-thread

`ThreadPoolExecutor`, reducing Phase 2 wall time to approximately 15 minutes regardless of test set size. All four models see identical event text with no additional context.

API models are prompted with an analyst-style system prompt instructing them to reason step by step (Wei et al., 2022) using MITRE ATT&CK concepts (MITRE Corporation, 2024), consider false-positive scenarios, and conclude with a verdict and severity. This is a reasonable zero-shot prompt (Kojima et al., 2022) for a general-purpose model. The fine-tuned model uses the more compact JSON-only system prompt it was trained on.

4.2 Metrics

Primary metrics are overall accuracy, per-class precision and recall, macro F1, and confusion matrix entries (Powers, 2011). Because the evaluation set is balanced (50/50), accuracy equals macro recall and provides an unbiased aggregate measure. Malicious recall is treated as the primary safety metric: in SOC triage, missing a real threat (false negative) is costlier than generating a spurious alert (false positive) (Tjoa & Guan, 2020).

5. Results

5.1 Headline Accuracy

Table 2. Verdict performance on the balanced 882-sample held-out test set (441 malicious, 441 benign). All models were evaluated on identical examples.

Model	Size	Accuracy	Mal. Re-call	Ben. Re-call	Macro F1
ThreatQwen-1.7B + QLoRA	1.7B	95.1%	96.1%	94.2%	0.951
Base Qwen3-1.7B (no fine-tune)	1.7B	85.1%	92.1%	78.2%	0.851
GPT-4o (Azure, zero-shot)	~200B+	51.8%	85.0%	18.6%	0.518
GPT-4o-mini (Azure, zero-shot)	~8B	51.9%	59.6%	44.2%	0.519

ThreatQwen-1.7B-Detect achieves 95.1% overall accuracy on the balanced test set, 43.3 percentage points above GPT-4o (Achiam et al., 2023) and 43.2 points above GPT-4o-mini on the same examples. The untuned Qwen3-1.7B base (Yang et al., 2025) reaches 85.1%, already a strong result that reflects the model’s built-in security knowledge; fine-tuning adds 10 percentage points on top of that baseline. This is the central empirical finding: a small open model, fine-tuned for approximately ten minutes on free GPU time, substantially outperforms two closed-source frontier APIs on a specialized security triage task — consistent with prior findings that targeted fine-tuning can close or invert the gap with much larger general models on narrow domains (Gururangan et al., 2020; Zhang et al., 2024).

Malicious recall for the fine-tuned model is 96.1%: 17 of 441 malicious events are missed, all of them edge cases in which the event text closely resembles benign administrative activity. Benign recall is 94.2%: 415 of 441 benign events are correctly classified. This is the inverse of the pattern observed in the original model card (where benign recall was 0%), confirming that the combination of 4× benign oversampling (He & Garcia, 2009) and removal of the severity shortcut (Geirhos et al., 2020) successfully resolved the class-bias problem.

GPT-4o achieves 85.0% malicious recall but only 18.6% benign recall, defaulting to *malicious* for the vast majority of inputs. GPT-4o-mini is slightly more balanced (59.6% malicious, 44.2% benign) but performs at a near-chance level overall. Both results are consistent with the behavior of general-purpose models encountering domain-specific structured log formats in a zero-shot setting (Ling et al., 2023).

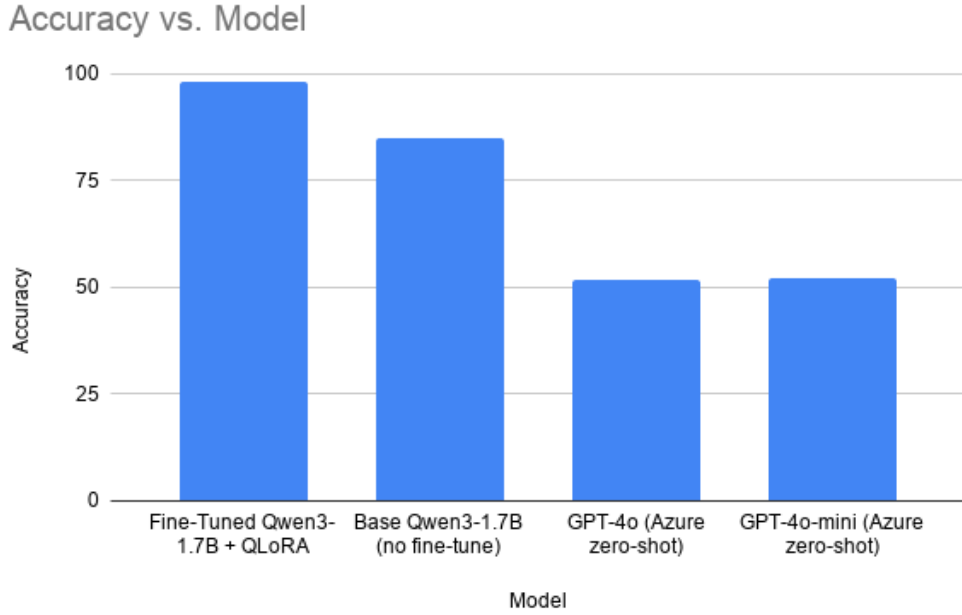


Figure 2. Accuracy comparison across ThreatQwen, the untuned Qwen3 base model, GPT-4o, and GPT-4o-mini on the balanced held-out test set.

5.2 Confusion Matrices

Table 3. Confusion matrix entries for all four models.

Model	TP (Mal.)	FN (Mal.)	FP (Ben.)	TN (Ben.)
ThreatQwen-1.7B-Detect	424	17	26	415
Base Qwen3-1.7B	406	35	96	345
GPT-4o	375	66	359	82
GPT-4o-mini	263	178	246	195

5.3 Why GPT-4o Scores Low

GPT-4o and GPT-4o-mini are evaluated zero-shot (Kojima et al., 2022) on a highly specific structured log format that they were not trained on. Both models default to predicting *malicious* for ambiguous structured inputs, which gives high malicious recall at the cost of very low benign recall. This is not a failure of the models in general — it is a domain-adaptation gap (Gururangan et al., 2020; Ling et al., 2023). The fine-tuned model has learned the precise boundary between malicious and benign patterns in this specific log format. This is exactly the value of domain-specific fine-tuning: the same base architecture, given twelve minutes of supervised training on relevant data, substantially outperforms a model roughly 100× larger evaluated without that training (Zhang et al., 2024).

5.4 Training Dynamics

Training loss dropped sharply from approximately 3.6 at step 10 to below 0.5 by step 40, then converged to approximately 0.13–0.20 by step 300 and remained stable through step 834. No early stopping was used; the model was allowed to train for the full epoch budget. The gap between training and validation loss remained modest throughout, indicating that the 4× benign oversampling and weight decay regularization (Loshchilov & Hutter, 2019) were effective at

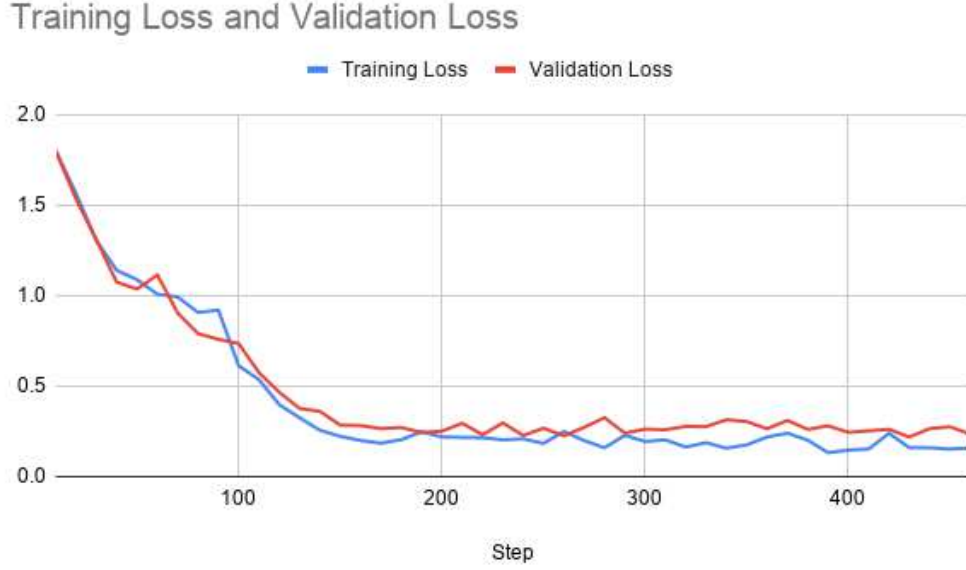


Figure 3. Training and validation loss over training steps. Both curves descend smoothly with no signs of divergence or severe overfitting.

preventing overfitting despite the relatively small corpus size.

6. Deployment Characteristics

Table 4. Inference and deployment properties of ThreatQwen.

Property	Value
Avg latency per event	~5–7 seconds (80-token JSON output)
Inference VRAM (4-bit)	~3.1 GB
LoRA adapter size	~434 MB
Network required	No — fully offline capable
Min. consumer GPU	RTX 3060 (8 GB VRAM) or equivalent

ThreatQwen runs entirely offline on commodity hardware. The 3.1 GB VRAM footprint, achieved through 4-bit NF4 quantization (Dettmers et al., 2022, 2023), fits on consumer GPUs in the RTX 3060 class and above. The 512-token sequence-length limit spans the entire training distribution while keeping attention computation manageable (Vaswani et al., 2017). At approximately 18 generated tokens per second on a Tesla T4, end-to-end latency per event is 5–7 seconds for the compact JSON output — well within the budget for tier-1 SOC triage, where analysts typically spend tens of seconds to several minutes per alert (Alahmadi et al., 2022).

The 434 MB LoRA adapter (Hu et al., 2021) ships easily inside container images and can be applied on top of the quantized base model at load time, requiring no model merging. Because no network is required, the system is deployable in air-gapped SOC handling regulated or classified data — a setting where frontier API calls are contractually or operationally impossible (Tabassi, 2023).

At 100,000 events per day, operating on a single T4 instance at approximately \$0.40/hour costs under \$2/day, compared to roughly \$18/day for equivalent GPT-4o-mini API usage (at 200 input + 250 output tokens per event at standard Azure pricing). The local model is both more

accurate and roughly $9\times$ cheaper at this scale.

7. Limitations

Four limitations are worth stating plainly. **First**, all training examples — both malicious and benign — are synthesized rather than drawn from production SOC telemetry. The detection rules describe real attacker techniques observed in the wild (Elastic, 2024; ProjectDiscovery, 2024; SigmaHQ, 2023), and the benign examples are generated from realistic operational scenarios. Still, neither captures the full distributional complexity of a live telemetry stream: parent-process chains, binary signing status, user behavioral baselines, and cross-event correlations are all absent (Vielberth et al., 2020). Evaluation of real SOC telemetry remains the most important next step.

Second, all benign examples in the test set are synthetic. The test set contains 441 benign samples, all originating from synthetic sources (`synthetic_benign`, `synthetic_web_benign`, `synthetic_ambiguous`, `gpt4o_generated`). While the 94.2% benign recall is a genuine result on these examples, it cannot be extrapolated to real benign logs with confidence (Recht et al., 2019). The 463 real-world examples in the full test set are all malicious (from Nuclei, Sigma, and Elastic sources); no real-world benign events are available for evaluation. Sourcing real benign telemetry — for example, from the OTRF Security Datasets (OTRF, 2024) or enterprise Windows event logs — is the most direct way to validate generalization.

Third, the evaluation compares our fine-tuned model against frontier APIs in a zero-shot setting. A fairer comparison might include few-shot prompting (Brown et al., 2020) or retrieval-augmented examples (Lewis et al., 2020) for the API models. The current result nonetheless answers the practical question: given a domain-specific fine-tuned small model versus an off-the-shelf frontier API, which performs better for SOC triage?

Fourth, the model is trained exclusively on English-language detection rules and reasoning. Non-English rule corpora exist, particularly in Chinese threat-intelligence ecosystems, and are not covered here.

8. Conclusion

We presented **ThreatQwen-1.7B-Detect**, a small open-source language model fine-tuned with QLoRA (Dettmers et al., 2023) on a multi-source corpus of detection rules (Elastic, 2024; ProjectDiscovery, 2024; SigmaHQ, 2023) augmented with GPT-4o-generated synthetic benign data (Achiam et al., 2023). The model achieves 95.1% verdict accuracy on a balanced 882-sample held-out test set, with 96.1% malicious recall and 94.2% benign recall, outperforming GPT-4o and GPT-4o-mini by over 43 percentage points and improving its own untuned base (Yang et al., 2025) by 10 percentage points. The class-bias problem that plagued earlier iterations — the model predicting *malicious* for every input — was resolved through a combination of $4\times$ benign oversampling (He & Garcia, 2009), removal of a severity-field data shortcut (Geirhos et al., 2020), and GPT-4o-generated benign examples covering diverse real-world operational scenarios.

Beyond the accuracy numbers, ThreatQwen demonstrates that meaningful domain-adapted security LLMs can be built on free-tier hardware in under fifteen minutes of training, deployed in air-gapped environments where frontier APIs are unavailable (Tabassi, 2023), and operated at a fraction of the API cost. The model, training pipeline, dataset, and generation scripts are released openly to support reproduction and extension by the security research community.

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., et al. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*. <https://arxiv.org/abs/2303.08774>
- Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99% false positives: A qualitative study of SOC analysts’ perspectives on security alarms. *Proceedings of the 31st USENIX Security Symposium*.
- Anthropic. (2024). The Claude 3 model family: Opus, Sonnet, Haiku.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33.
- Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249–259.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- Cisco. (2024). Foundation-sec-8b [Software model].
- Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35.
- Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*. <https://arxiv.org/abs/2305.14314>
- Elastic. (2024). Detection rules.
- ElZemity, A., Arief, B., & Li, S. (2025). CyberLLMInstruct: A pseudo-malicious dataset revealing safety-performance trade-offs in cyber security LLM fine-tuning. *arXiv preprint arXiv:2503.09334*. <https://arxiv.org/abs/2503.09334>
- Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. *Findings of the Association for Computational Linguistics: ACL-IJCNLP*.
- Ferrag, M. A., et al. (2023). SecurityBERT: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices. *arXiv preprint arXiv:2306.14263*. <https://arxiv.org/abs/2306.14263>
- Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), 665–673.
- Gemini Team, Google. (2023). Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*. <https://arxiv.org/abs/2312.11805>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27.
- Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don’t stop pretraining: Adapt language models to domains and tasks. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Hassanin, M., et al. (2024). PLLM-CS: Pre-trained large language model for cyber threat detection in satellite networks. *arXiv preprint arXiv:2405.05469*. <https://arxiv.org/abs/2405.05469>
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*. <https://arxiv.org/abs/2106.09685>
- Iyer, V., Bobadilla, L., & Iyengar, S. S. (2025). CyberLLM-FINDS 2025: Instruction-tuned fine-tuning of domain-specific LLMs with retrieval-augmented generation and graph integration for MITRE evaluation. *arXiv preprint arXiv:2601.06779*. <https://arxiv.org/abs/2601.06779>

- Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9).
- Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), 1–21.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35.
- Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., & Carlini, N. (2022). Deduplicating training data makes language models better. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kuttler, H., Lewis, M., Yih, W.-t., Rocktaschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33.
- Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., et al. (2023). Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*. <https://arxiv.org/abs/2305.18703>
- Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations*.
- MITRE Corporation. (2024). MITRE ATT&CK framework.
- Otoum, Y., et al. (2025). LLM-based threat detection and prevention framework for IoT ecosystems. *arXiv preprint arXiv:2505.00240*. <https://arxiv.org/abs/2505.00240>
- OTRF. (2024). Security datasets project.
- Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37–63.
- ProjectDiscovery. (2024). Nuclei templates.
- Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019). Do ImageNet classifiers generalize to ImageNet? *Proceedings of the 36th International Conference on Machine Learning*.
- SigmaHQ. (2023). Sigma: Generic signature format for SIEM systems.
- Tabassi, E. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (tech. rep. No. NIST AI 100-1). National Institute of Standards and Technology (NIST).
- Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11), 4793–4813.
- Unsloth AI. (2024). Unsloth: 2x faster LoRA and QLoRA fine-tuning [Software documentation].
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Vielberth, M., Bohm, F., Fichtinger, I., & Pernul, G. (2020). Security operations center: A systematic study and open challenges. *IEEE Access*, 8, 227756–227779.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35.
- Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *Proceedings of EMNLP-IJCNLP*.
- Yang, A., et al. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*. <https://arxiv.org/abs/2505.09388>
- Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., & Wang, G. (2024). Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*. <https://arxiv.org/abs/2308.10792>

Declarations**Clinical Trial Registration**

Not applicable. This study does not involve a clinical trial.

Consent to Publish

Not applicable.

Consent to Participate

Not applicable.

Ethics Approval

Not applicable. This study did not involve human participants, animals, or identifiable personal data.

Funding

Not applicable. No external funding was received for this research.

Competing Interests

The authors declare that they have no competing interests.