International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Generative AI for Next-Generation Cyber Security: From Threat Intelligence to Autonomous Defence

Rishabh Pandey
Department of Computer Science and Information Technology
Babu Banarasi Das University, Lucknow, India
rishabh.pandey1806@gmail.com

Sahil Ali
Department of Computer Science and Information Technology
Babu Banarasi Das University, Lucknow, India
alisahil2304@gmail.com

Shivam Gupta
Department of Computer Science and Information Technology
Babu Banarasi Das University, Lucknow, India
sg1043998@gmail.com

Mrs.Deepti Gupta
Assistant Professor
Department of Computer Science and Information Technology
Babu Banarasi Das University, Lucknow, India
deepti.gupta.16@gmail.com

***Abstract*—Cyber defence is being stretched by two forces at once: an expanding attack surface and adversaries who now use generativeAI models to write phishing lures, mutate malware and automate reconnaissance. The same family of models, which includes large language models (LLMs), autoencoders and generative adversarial networks, can also help defenders, yet recent surveys examine threat intelligence, detection and autonomous response in isolation. This paper follows a single piece of intelligence from ingestion to action. We review work published since 2024 and propose GenDefend, a five-layer reference framework that links generative threat-intelligence extraction, generative detection, LLM-based reasoning and a risk-gated autonomous response layer. To test parts of the design we built a reproducible synthetic flow dataset of 12,000 records with six traffic classes and ran controlled experiments. Autoencoder-based latent augmentation raised the macro recall of a random forest from 0.850 to 0.862, removing threat-intelligence features cost 2.8 points of macro F1, and a reconstruction-based scorer reached AUC values of 0.89–0.96 on attack types it had never seen. A Monte-Carlo simulation under stated assumptions suggests that graded autonomy could shorten mean time to containment several-fold. We close with the risks that must be managed before such systems are trusted in production.**

***Index Terms*—Autonomous defence, cyber threat intelligence, cybersecurity, generative AI, large language models.**

I. Introduction

Security teams are asked to defend a perimeter that no longer has edges. Cloud workloads, remote staff, connected devices and third-party code have widened the attack surface faster than most organisations can recruit analysts to watch it. At the same time, attackers have begun borrowing the machine-learning tools that defenders rely on, which shortens the gap between a new weakness becoming known and a working campaign appearing in the wild [1].

Generative AI (GenAI) changes the balance in both directions. The term covers large language models (LLMs), generative adversarial networks (GANs), variational autoencoders (VAEs) and diffusion models. For a defender, such models can read thousands of unstructured threat reports a day, produce realistic examples of rare attacks for training, and draft response steps in plain language [2], [3]. For an attacker, the same models lower the skill needed to produce convincing phishing, polymorphic malware and deepfake-assisted fraud [1], [4].

Existing surveys tend to look at one slice of this picture. Some concentrate on cyber threat intelligence (CTI) [2]; others map LLM use across security tasks [3], [5]; others study autonomous platforms such as drones and vehicles [6]. What is still missing is an end-to-end view that follows a piece of intelligence from the moment it is ingested to the moment a control on the network actually changes, and that asks when that last step should be left to a machine.

This paper takes that end-to-end view. Our contributions are fourfold. (i) We organise recent work (2024 onwards) along a pipeline of intelligence, detection, reasoning and response. (ii) We propose GenDefend, a five-layer reference framework in which the generative components feed one another and every automated action must pass a risk-aware policy gate. (iii) We build a reproducible synthetic flow dataset and use it to measure what generative augmentation, threat-intelligence features and reconstruction-based scoring each add to detection. (iv) We use a Monte-Carlo simulation to explore how the level of autonomy affects containment time, and we discuss the risks that come with it.

Section II reviews related work, Section III describes the framework, Section IV the experimental set-up, Section V the results, and Section VI the open challenges before Section VII concludes.

II. Related Work

A. Generative Models in Security

Each family of generative model has found a different job in security. GANs are used to synthesise attack traffic so that intrusion detectors can be trained on balanced data and stress-tested against evasive samples. Autoencoders and VAEs learn what normal behaviour looks like and flag whatever they cannot reconstruct. Transformers and LLMs read text and code, which makes them natural tools for reports, logs and malware analysis. Andreoni et al. [6] sort these families by how they help autonomous systems, covering anomaly detection, predictive maintenance and adaptive threat response, and they stress that resilience requires more than accuracy. Domain-adapted language models such as CySecBERT [7] and lightweight, privacy-preserving transformers for IoT traffic [8] show that smaller specialised models can be practical where large general ones are not.

B. Generative AI for Threat Intelligence

Balasubramanian et al. [2] survey the data sources that feed CTI, namely reports, blogs, social media, network traffic,

malware samples and dark-web content, and they connect each to a generative technique. Work on open-source intelligence is moving in the same direction: Shafee et al. [9] compare chatbots on OSINT-based threat-awareness questions, while Sufi [10] mines historical incident reports with a GPT-based pipeline. Benchmarks such as CyberMetric [11] give a way to check whether a model actually knows enough security to be trusted with these tasks.

C. Autonomy and Its Risks

Automation cuts both ways. PentestGPT [12] shows that LLM agents can carry out multi-step penetration-testing work, which is equally useful to defenders and attackers. Yao et al. [4] describe LLMs as defensive tools, as attack tools, and as vulnerable targets in their own right, and jailbreak studies [13] confirm that guardrails can be talked around. Architectural answers are starting to appear, such as isolating the execution of LLM-driven apps [14], and the robotics literature on variable autonomy [15] offers a vocabulary for deciding when a human must stay in the loop.

D. Positioning of This Work

Table I places this paper next to the closest recent studies. As far as we can tell from their abstracts, none of them links the intelligence, detection and response stages in a single design that is also exercised by experiments.

TABLE I
COVERAGE OF CLOSELY RELATED RECENT STUDIES

Ref.	Year	Main emphasis	CTI	Det.	Resp.	Method
[1]	2024	Offensive misuse	–	○	–	Case analysis
[2]	2025	GenAI for CTI	●	○	–	Survey, cases
[3]	2024	LLM apps and flaws	○	●	○	Review
[5]	2024	LLMs for security	○	●	○	Literature review
[6]	2024	Autonomous systems	–	●	○	Survey
Ours	2025	Intel. to response	●	●	●	Framework, tests

● main focus, ○ touched on, – not covered (our reading).

III. PROPOSED FRAMEWORK: GENDEFEND

A. Overview and Design Principles

GenDefend is organised as five layers (Fig. 1). Data flow downward, and a feedback path carries analyst decisions and incident outcomes back up so that every layer can be re-tuned. Four principles guided the design. First, intelligence should be enriched step by step rather than consumed raw. Second, whatever a generative model produces is advisory until it has been checked against another source or rule. Third, every automated action must be bounded, reversible and logged. Fourth, the degree of autonomy is a setting, not a switch, and it depends on risk.

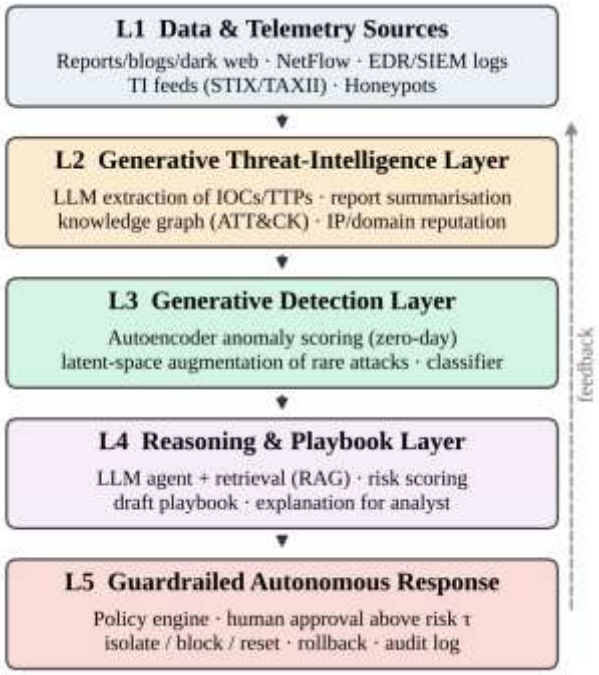


Fig. 1. The GenDefend reference architecture. Solid arrows show the forward flow of intelligence; the dashed arrow is the feedback path.

B. Generative Threat-Intelligence Layer (L2)

This layer turns messy text into structured, machine-usable knowledge. An LLM reads reports, advisories and forum posts and extracts indicators of compromise, ATT&CK techniques and named actors into a fixed schema (STIX-like JSON). Output is constrained to that schema and grounded by retrieval from a curated knowledge graph, which reduces the freedom the model has to invent entities. An indicator is promoted to the live blocklist only when two independent sources agree or an analyst approves it; the layer then publishes a confidence value c and a reputation score for each address and domain. In our prototype these appear as two numeric features, *ip_reputation* and *geo_risk*.

C. Generative Detection Layer (L3)

Detection runs along two paths. A supervised classifier handles attack classes that have labelled examples. Because rare classes such as data exfiltration are exactly the ones that matter and exactly the ones that are scarce, we enlarge them with a generative step. For a minority class c we train an autoencoder with encoder f_θ and decoder g_φ by minimising

$$\mathcal{L}_{AE} = \frac{1}{N_c}\sum_{i=1}^{N_c}\left\|x_i - g_\varphi\left(f_\theta(x_i)\right)\right\|_2^2 \qquad (1)$$

fit a Gaussian $N(\mu_c, \Sigma_c)$ to the latent codes, draw new codes with the covariance inflated by $\lambda = 1.6$ for diversity, and decode them. Each decoded sample is averaged with a jittered copy of a real record (noise $\sigma = 0.15$) so that it stays near the real data manifold. The second path is for attacks never seen in training. An autoencoder trained on benign traffic only produces a reconstruction error $e(x)$ for each flow, and this is fused with the threat-intelligence reputation $r(x)$ after rank normalisation:

$$s(x) = 0.7\,\mathrm{rank}[e(x)] + 0.3\,\mathrm{rank}[r(x)] \qquad (2)$$

D. Reasoning and Playbook Layer (L4)

Once a flow is flagged, an LLM agent gathers context (asset owner, business criticality κ, recent alerts, matching ATT&CK techniques) through retrieval, explains in plain language why the alert fired, and drafts a response playbook. We

combine the detector probability p with the threat-intelligence confidence c into a single action confidence

$$q(x) = \omega\, p(x) + (1 - \omega)\, c(x), \quad \omega = 0.7 \qquad (3)$$

E. Guardrailed Autonomous Response (L5)

Playbook steps are classified as reversible (rate-limiting, a time-boxed IP block, quarantining a VLAN, forcing an MFA reset) or irreversible (wiping a host, deleting an account). A step is executed automatically only if $q \geq \tau_a$, the step is reversible, and the asset criticality is below a limit κ_{max}. Everything else is routed to an analyst with the explanation attached. Automatic actions carry an expiry timer and a rollback, are written to an immutable audit log, and can be disabled by a global kill switch. The LLM never holds execution credentials; it can only submit a request to the policy engine, in the spirit of isolation architectures such as SecGPT [14]. Algorithm 1 summarises the full decision loop.

Algorithm 1: GenDefend decision loop

1. Ingest reports; L2 extracts indicators and promotes one only if ≥ 2 sources agree or an analyst approves.
2. For each flow x: build features, including reputation $r(x)$.
3. $p \leftarrow 1 - \mathrm{P}(\text{benign} \mid x)$ from the classifier; $e \leftarrow$ autoencoder reconstruction error.
4. If p is low but $s(x)$ is high, label the flow as a suspected unknown attack.
5. $q \leftarrow \omega p + (1 - \omega)c$; L4 gathers context and drafts playbook P.
6. For each step a in P: if $q \geq \tau_a$, a is reversible and $\kappa \leq \kappa_{max}$, execute it with an expiry timer and an audit entry; otherwise queue it for analyst approval.
7. Store the outcome; periodically re-fit the augmenter and re-tune τ_a.

F. Threat Model

We assume a mid-sized enterprise defender with a SIEM, flow telemetry and an external threat-intelligence subscription. The adversary may use GenAI to craft lures and mutate malware, may try to evade the detector by shaping traffic, may plant misleading content in public sources that the L2 layer reads, and may inject instructions into log fields or documents that reach the LLM. We assume the adversary cannot directly modify the policy engine, the audit log or the training pipeline. The design is meant to degrade gracefully when these assumptions are partly violated: a poisoned indicator needs a second source to be promoted, and a confused LLM can only propose actions, never execute them.

IV. Experimental Set-up

A. Dataset

We could not rely on a public corpus because we also needed a threat-intelligence field next to the usual flow statistics. We therefore generated *GenDefend-Flow*, a synthetic enterprise dataset of 12,000 flow records with 16 features and six classes. The features cover volume (bytes, packets, duration), timing (inter-arrival mean and deviation), protocol behaviour (SYN ratio, payload entropy, TLS anomalies, DNS rate), identity behaviour (failed logins, distinct destination ports, off-hours activity) and threat intelligence (IP reputation and geographic risk). Class-conditional distributions deliberately overlap, the classes are imbalanced, and 4% of labels are flipped at random to imitate imperfect labelling (Table II). Because the data are synthetic, absolute scores should not be read as real-world performance.

TABLE II
Composition of the GenDefend-Flow Dataset

Class	Records	Share	Characteristic behaviour
Benign	7,440	62.0%	Mixed web, mail, DNS
DoS	1,431	11.9%	High packets, SYN-heavy
PortScan	1,236	10.3%	Many ports, short flows
BruteForce	944	7.9%	Failed logins, off-hours
PhishC2	587	4.9%	High DNS rate, TLS oddities
Exfiltration	362	3.0%	Large uploads, high entropy

B. Models, Protocol and Metrics

Five detectors are compared: logistic regression (LR), a 200-tree random forest (RF), a two-hidden-layer MLP (64–32), RF trained after simple interpolation-based oversampling (a SMOTE-like baseline), and the GenDefend variants in which BruteForce, PhishC2 and Exfiltration are enlarged by the autoencoder method of Section III-C to the size of the PortScan class. The autoencoder has a 24–4–24 tanh bottleneck. We use a stratified 70/30 split, repeat with three random seeds, and report mean ± standard deviation. Metrics are accuracy, macro precision, macro recall, macro F1 and per-class F1; for unknown-attack detection we use ROC AUC. For that test, the autoencoder is trained on 60% of the benign records alone, and each attack class is held out in turn and mixed with unseen benign flows at a 1:3 ratio. Isolation Forest, a one-class SVM and a plain autoencoder (no intelligence term) serve as comparators.

C. Response Simulation

Layers L4 and L5 were not implemented with a live LLM; instead we model their effect on incident handling. For each of four set-ups we draw 4,000 incidents with log-normal detection and response times. The set-ups are a manual SOC (recall 0.78, median detection 42 min, no automation), a rule-based SOAR (0.83, 18 min, 55% of incidents handled automatically), GenDefend with a human in the loop (0.94, 4.5 min, 80% automatic) and GenDefend fully automatic (0.94, 4.5 min, 100%). Missed incidents are discovered after a median of six hours. **These parameters are assumptions, not measurements**; the simulation therefore shows how sensitive containment time is to autonomy level, not what a real deployment would achieve.

V. Results and Discussion

A. Detection of Known Attacks

TABLE III
Known-Attack Detection (Mean ± Std, Three Seeds)

Model	Accuracy	Macro-P	Macro-R	Macro-F1
LR	0.950±.002	**0.922±.007**	0.873±.009	**0.895±.006**
RF	0.947±.003	0.919±.005	0.850±.016	0.878±.010
MLP	0.946±.003	0.893±.001	0.877±.016	0.884±.009
RF + interpolation	**0.951±.001**	0.913±.004	**0.879±.009**	0.895±.004
GenDefend (AE + RF)	0.949±.002	0.913±.003	0.862±.015	0.885±.008
GenDefend (AE + MLP)	0.942±.001	0.878±.003	0.878±.015	0.877±.006

Best value per column in bold.

Table III shows that all models reach about 94–95% accuracy, and the headline numbers are close. Accuracy is a weak guide here because benign flows make up over 60% of the data, so we focus on macro recall and per-class F1. Autoencoder augmentation lifts the random forest's macro recall from 0.850 to 0.862 and its macro F1 from 0.878 to 0.885, at a small cost in precision. The gain is concentrated where it should be (Fig. 2): F1 on exfiltration rises from 0.627 to 0.659. We must

be candid about two things, however. The simple interpolation baseline does at least as well (macro F1 0.895, exfiltration F1 0.710), so our experiment does not show an advantage for the autoencoder over cheaper oversampling. And plain logistic regression (0.895) is just as strong as the ensemble, which is probably a side-effect of the near-Gaussian structure of synthetic features. We expect the picture to change on real traffic, where feature dependencies are far less regular.

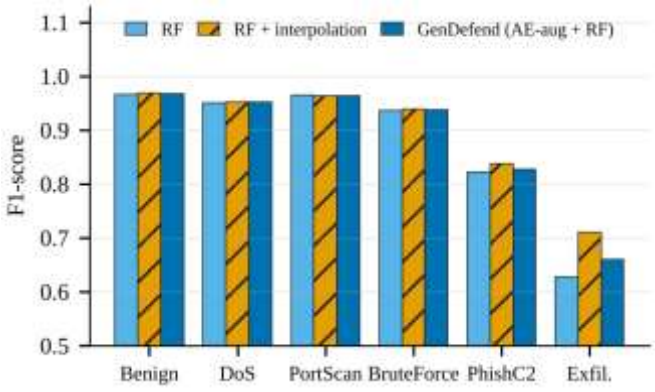


Fig. 2. Per-class F1 of the random forest with no augmentation, with interpolation oversampling, and with autoencoder augmentation (mean of three seeds).

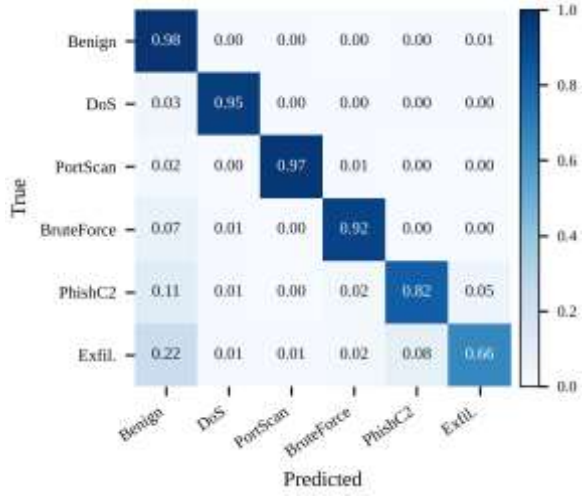


Fig. 3. Row-normalised confusion matrix of GenDefend (AE + RF) on one test split. Most errors occur between exfiltration, PhishC2 and benign traffic.

The confusion matrix in Fig. 3 explains where the remaining errors lie. Volumetric and scan-type attacks are almost always recognised. The hard cases are low-and-slow behaviours that resemble ordinary encrypted traffic: exfiltration is most often mistaken for benign flows or for command-and-control activity, which is consistent with how such attacks hide in practice.

B. Contribution of Each Component

TABLE IV
ABLATION OF THE DETECTION PIPELINE (RF CLASSIFIER)

Variant	Macro-F1	Rare-class F1
Full GenDefend	0.885±.008	0.808
w/o generative augmentation	0.878±.010	0.795
w/o TI features (rep, geo)	0.857±.010	0.758
w/o behavioural features	0.717±.004	0.563

Rare-class F1 is the mean F1 of BruteForce, PhishC2 and Exfiltration.

Table IV isolates each component. Removing the two threat-intelligence features lowers macro F1 by 2.8 points and rare-class F1 by 5.0 points, so the intelligence layer is doing real work in the pipeline. Removing the behavioural features hurts far more (0.717), which is a reminder that intelligence supplements, rather than replaces, behavioural evidence. Removing augmentation costs only 0.7 points of macro F1, a difference of the same order as the seed-to-seed deviation, so on this dataset it is the smallest of the three contributions.

Feature importances from the forest (Fig. 4) tell a consistent story. Reputation ranks number 1 out of 16 features (importance 0.141), followed by failed logins, distinct destination ports and DNS rate. A caution applies: we generated the reputation field so that malicious hosts score higher on average, so its prominence is partly built in. Real feeds are noisier, which is why we treat the reported benefit as an upper-bound indication.

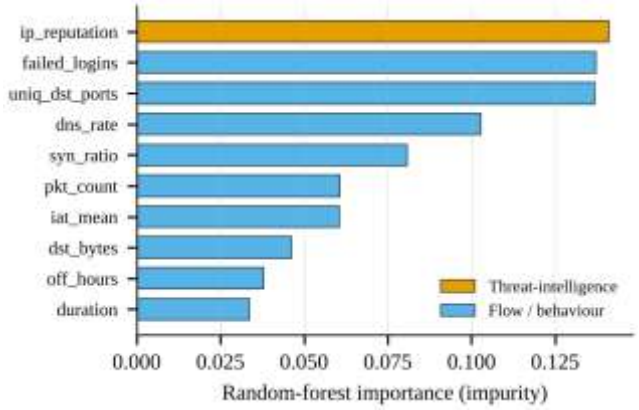


Fig. 4. Ten most important features of the random forest. Threat-intelligence features are highlighted.

C. Detection of Unseen Attacks

TABLE V
UNKNOWN-ATTACK DETECTION (ROC AUC, HELD-OUT CLASS)

Held-out class	IForest	OC-SVM	AE	GenDefend
PhishC2	0.943	0.955	0.948	**0.965**
Exfiltration	**0.904**	0.888	0.831	0.887
BruteForce	0.949	**0.971**	0.909	0.936

Mean of three seeds; best per row in bold.

Table V reports leave-one-attack-out results. Fusing reconstruction error with the reputation score is best for PhishC2 (AUC 0.965) and clearly improves on the plain autoencoder in all three cases. It is not uniformly best, though: for exfiltration Isolation Forest is ahead (0.904 against 0.887, Fig. 5) and for brute force the one-class SVM leads (0.971). A single unsupervised detector is therefore unlikely to be enough, and an ensemble of complementary scorers is a natural next step.

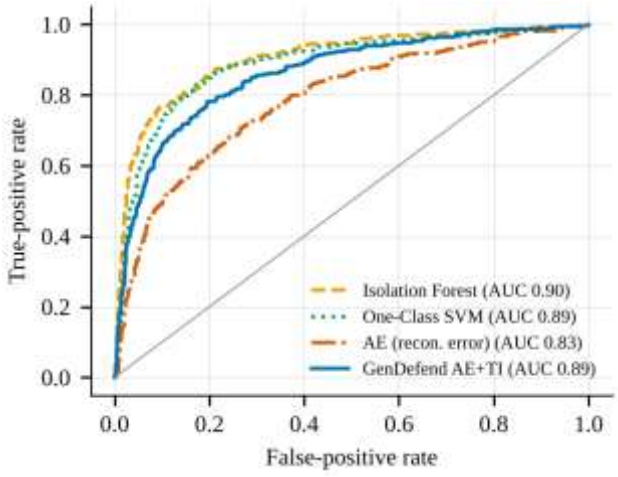


Fig. 5. ROC curves for exfiltration treated as an unknown attack (one seed).

D. Effect of Autonomy on Containment Time

TABLE VI
SIMULATED INCIDENT HANDLING (4,000 INCIDENTS PER SET-UP)

Set-up	MTTD	MTTR	Med.	≤60 min	Disrupt.
Manual SOC	124	235	165	1%	0.0%
Rule-based SOAR	87	168	88	27%	3.4%
GD (human-in-loop)	28	64	16	78%	1.7%
GD (fully auto)	26	36	9	95%	3.9%

Times in minutes. MTTD/MTTR: mean time to detect/respond. Disrupt.: share of incidents with an unnecessary containment. GD = GenDefend.

Under the assumptions of Section IV-C, mean time to respond falls from 235 min for the manual SOC to 64 min with human-in-the-loop GenDefend and 36 min when every step is automatic (Table VI, Fig. 6). The share of incidents contained within an hour rises from 1% to 78% and 95%. The price of full automation is visible too: unnecessary containments more than double (1.7% against 3.9%). Human approval above a risk threshold thus buys back most of the safety at the cost of some speed, which is the trade-off the L5 policy gate is meant to expose to operators.

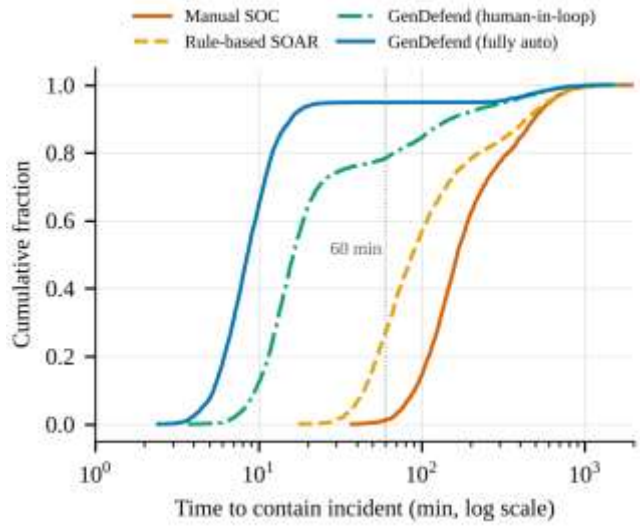


Fig. 6. Cumulative distribution of simulated time to contain an incident for the four set-ups (log scale).

E. Choosing the Autonomy Threshold

The threshold τ_a in Section III-E decides how much work is left to humans. Using the classifier probabilities from the first split and the reputation feature as a stand-in for c, we swept τ_a over the fused confidence q. Fig. 7 shows the expected shape. At $\tau_a = 0.6$, 93% of attack flows would be actioned automatically while 2.0% of benign flows would be actioned wrongly; at 0.7 the figures are 84% and 1.3%, and at 0.8 automation falls to 47% of attacks with only 0.6% false actions. A deployment would pick a point in the shaded band for reversible actions and a stricter one for anything that is costly to undo.

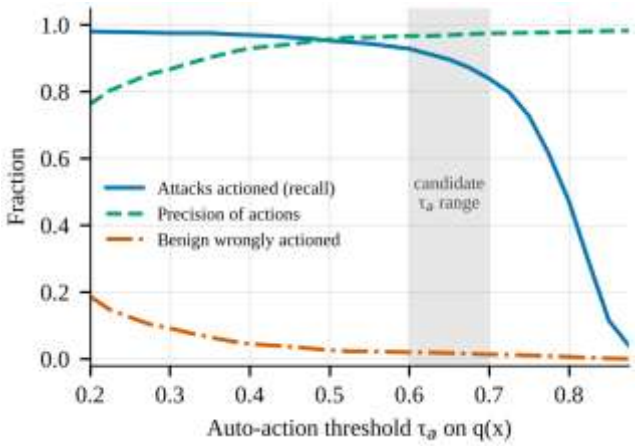


Fig. 7. Effect of the auto-action threshold on coverage and errors (one test split).

F. Computational Cost

Table VII lists what the prototype costs to run on a small two-core cloud container without a GPU. Training the whole supervised path takes a few seconds and scoring is fast enough for flow-level screening, so the generative detection layer is unlikely to be the bottleneck; the LLM calls in L2 and L4 would be, which is the reason for the tiered design discussed in Section VI.

TABLE VII
MEASURED COST OF THE PROTOTYPE (2 VCPU, NO GPU)

Component	Setting	Cost
Latent augmentation	1,270 synthetic records	0.9 s
RF training	200 trees, augmented set	3.8 s
RF inference	per 1,000 flows	24 ms
AE training	benign flows only	1.6 s
AE scoring	per 1,000 flows	0.6 ms

G. Threats to Validity

Three limits deserve emphasis. The dataset is synthetic and its classes were designed by us, so the numbers show relative behaviour of components, not field performance. The LLM-driven layers (L2 and L4) are specified but were not evaluated with a live model. And the simulation rests on assumed parameters. Validating the pipeline on real, labelled traffic and with a real LLM in the loop is the most important item of future work.

VI. CHALLENGES AND FUTURE DIRECTIONS

TABLE VIII
DUAL-USE VIEW OF GENERATIVE AI IN SECURITY

Area	Attacker use	Defender use
Social engineering	Fluent, targeted lures	Lure detection, user training
Malware	Variant generation	Variant synthesis for training
Reconnaissance	Automated OSINT	OSINT-driven CTI [9]
Vulnerability work	Exploit drafting	Automated testing [12]
Operations	Evasive scripting	Playbooks, triage

Qualitative summary drawn from the literature cited in this paper.

Dual use. Every capability described here is available to attackers too [1], [4], as Table VIII summarises. Defensive systems should be assumed to face adversaries that also use GenAI, and should be red-teamed accordingly. **Reliability of generated intelligence.** LLMs can fabricate indicators or misattribute campaigns; schema constraints, retrieval grounding and the two-source promotion rule of Section III-B reduce but do not remove this risk, and benchmarks such as CyberMetric [11] should be used to qualify a model before it is

trusted. **Attacks on the defender.** Prompt injection through crafted log lines or reports, data poisoning of training streams and jailbreaks [13] can turn an assistant against its owner, which is why we keep execution credentials away from the model [14].

Privacy and data governance. Security telemetry contains personal and commercial data. On-premise or privacy-preserving models [8] and careful retention rules are needed before logs are sent to external services. **Accountability.** When an automated step blocks a legitimate user, someone must be able to see why. Explanations, audit logs and variable-autonomy designs [15] are therefore requirements, not extras, and emerging policy work [16] suggests that regulators will expect them. **Cost and latency.** Large models are expensive to run on every alert; a tiered design in which cheap detectors filter and the LLM is consulted only for escalated cases is more realistic.

Our next steps are to repeat the experiments on public intrusion datasets and on live telemetry, to replace the LLM-dependent parts of L2 and L4 with an evaluated open-weight LLM, to compare autoencoders with GANs and diffusion models for augmentation, and to run a controlled study with analysts to calibrate the thresholds τ_a and κ_{max}.

VII. Conclusion

We examined how generative AI can support a defender from the first piece of intelligence to the final containment action, and proposed GenDefend as a five-layer design that keeps humans in control of high-risk decisions. Our experiments on a synthetic dataset indicate that threat-intelligence features add measurable value, that generative augmentation gives a modest gain for rare attacks without clearly beating simpler oversampling, and that reconstruction-based scoring is a useful but not universally best detector of unseen attacks. A simulation under stated assumptions suggests that graded autonomy can shorten containment considerably while exposing a clear trade-off with unnecessary disruption. Because the evidence is synthetic, the main value of this work is the structure it offers and the questions it sharpens; confirming the findings on real data is the next step.

Acknowledgment

The authors thank their guide, Mrs. Deepti Gupta, for her guidance and encouragement throughout this work.

References

[1] S. Metta, I. Chang, J. Parker, M. P. Roman, and A. F. Ehuan, "Generative AI in cybersecurity," 2024, *arXiv:2405.01674*.

[2] P. Balasubramanian, S. Liyana, H. Sankaran, S. Sivaramakrishnan, S. Pusuluri, S. Pirttikangas, and E. Peltonen, "Generative AI for cyber threat intelligence: Applications, challenges, and analysis of real-world case studies," *Artif. Intell. Rev.*, vol. 58, Art. no. 336, 2025, doi: 10.1007/s10462-025-11338-z.

[3] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, and N. Tihanyi, "Generative AI and large language models for cyber security: All insights you need," 2024, *arXiv:2405.12750*.

[4] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *High-Confidence Comput.*, vol. 4, no. 2, 2024.

[5] H. Xu *et al.*, "Large language models for cyber security: A systematic literature review," 2024, *arXiv:2405.04760*.

[6] M. Andreoni, W. T. Lunardi, G. Lawton, and S. Thakkar, "Enhancing autonomous system security and resilience with generative AI: A comprehensive survey," *IEEE Access*, vol. 12, pp. 109470–109493, 2024, doi: 10.1109/ACCESS.2024.3439363.

[7] M. Bayer, P. Kuehn, R. Shanehsaz, and C. Reuter, "CySecBERT: A domain-adapted language model for the cybersecurity domain," *ACM Trans. Privacy Secur.*, vol. 27, no. 2, pp. 1–20, May 2024.

[8] M. A. Ferrag *et al.*, "Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices," *IEEE Access*, vol. 12, pp. 23733–23750, 2024.

[9] S. Shafee, A. Bessani, and P. M. Ferreira, "Evaluation of LLM chatbots for OSINT-based cyber threat awareness," 2024, *arXiv:2401.15127*.

[10] F. Sufi, "An innovative GPT-based open-source intelligence using historical cyber incident reports," *Natural Lang. Process. J.*, vol. 7, Art. no. 100074, Jun. 2024.

[11] N. Tihanyi, M. A. Ferrag, R. Jain, T. Bisztray, and M. Debbah, "CyberMetric: A benchmark dataset based on retrieval-augmented generation for evaluating LLMs in cybersecurity knowledge," 2024, *arXiv:2402.07688*.

[12] G. Deng *et al.*, "PentestGPT: Evaluating and harnessing large language models for automated penetration testing," in *Proc. 33rd USENIX Secur. Symp.*, 2024.

[13] H. Jin *et al.*, "JailbreakZoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models," 2024, *arXiv:2407.01599*.

[14] Y. Wu, F. Roesner, T. Kohno, N. Zhang, and U. Iqbal, "SecGPT: An execution isolation architecture for LLM-based systems," 2024, *arXiv:2403.04960*.

[15] L. Methnani, M. Chiou, V. Dignum, and A. Theodorou, "Who's in charge here? A survey on trustworthy AI in variable autonomy robotic systems," *ACM Comput. Surv.*, vol. 56, no. 7, pp. 1–32, Jul. 2024.

[16] M. Christodorescu *et al.*, "Securing the future of GenAI: Policy and technology," *Cryptol. ePrint Arch.*, Paper 2024/855, 2024.