WHITEPAPER

Choosing the Right Generative Model

LLMs are only PART of a more sophisticated approach to threat detection

By Leigh Shlomovich

Executive Summary

The term “generative AI” has become synonymous with large language models (LLMs). In cybersecurity, this conflation is costly. Security teams are being sold language-processing tools for a fundamentally non-linguistic problem: detecting anomalous human behaviour across numerical, temporal, and stateful data streams.

This paper presents three core ideas:

- Generative models predate ChatGPT by decades and describe a broad statistical family, not just LLMs.
- Security data is not just linguistic, it's also behavioural, numerical, temporally dynamic and contextually dependent. Forcing it through an LLM’s language bottleneck introduces three layers of lossy abstraction.
- Purpose-built generative models (including HMMs, GMMs, VAEs and Naive Bayes) directly learn the probability distributions that matter for detection, delivering interpretable, auditable, and statistically grounded results.

Key stat: *According to the SANS 2024 SOC Survey, 66% of SOC teams report they cannot keep pace with the volume of alerts they receive. The root cause is not a lack of AI, it is the wrong kind of AI.*

1. Introduction: The Generative Misnomer

In 2023, “generative AI” entered the mainstream lexicon. With it came an assumption that would yet again distort terminology across the security industry: that the word “generative” is synonymous with “LLM.” It is not.

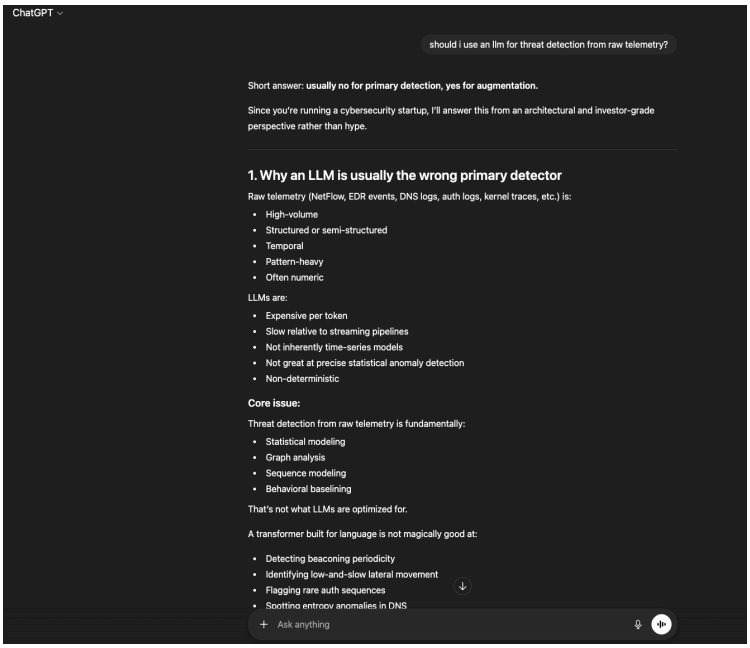
The statistical concept of a generative model is over 200 years old. It refers to any probabilistic model that learns the underlying distribution of data — P(X) or P(X, Y) — rather than simply learning to discriminate between categories. An LLM is one implementation of this framework, trained on the joint distribution of words in text through a vast training corpus. But a Hidden Markov Model trained on authentication sequences, a Gaussian Mixture Model profiling network traffic, and a Variational Autoencoder detecting lateral movement are equally, rigorously generative, and often far more appropriate for the task.

As MIT's explainer on generative AI notes [Zewe A, MIT News, Explained: Generative AI], the lineage of these models traces back to Andrey Markov's 1906 work on random processes, long before neural networks, let alone modern transformer architecture. A 2001 paper by Andrew Ng and Michael Jordan formally compared the statistical properties of generative versus discriminative classifiers [Ng. A. Y, Jordan, M. I (2001)], a conversation that continues to evolve and remain deeply relevant to model selection in security today.

Ng, A. Y., & Jordan, M. I. (2001). On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. NIPS '01.

MIT News (2023). Explained: Generative AI. https://news.mit.edu/2023/explained-generative-ai-1109 (Accessed 25.02.26)

This paper is directed at CISOs, SOC architects, and security platform buyers who are navigating an AI landscape flooded with LLM-based solutions. For clarity, LLMs are not without value in cybersecurity, far from it, they demonstrably assist with log summarisation, threat intelligence insights from unstructured news feeds, and analyst augmentation. However, in this paper we hope to highlight and restate the importance of "right tool for the right job". For the core problem of real-time anomalous behaviour detection, LLMs are the wrong generative model, and the industry's uncritical adoption of them introduces measurable statistical inefficiency into the detection pipeline.



Core message: *Security needs models that ask "what actions follow other actions given system state?" LLMs are built to answer "what words follow other words?" These are different probability distributions. Collapsing the former into the latter loses temporal ordering, numerical precision, state dependency, and causal chain integrity.*

2. What "Generative" Actually Means

2.1 The Statistical Foundation

In statistical theory, models can be divided into two families based on what they learn from data. Discriminative models learn P(Y|X). This is the conditional probability of a label given inputs. Generative models learn P(X, Y). This is the joint distribution of both inputs and labels. The practical difference is significant: **generative models capture the data-generating process** itself, enabling them to detect anomalies, generate synthetic samples, and provide calibrated probabilistic scores. All of this is enabled because we've captured what the data looks like.

To make this concrete: a discriminative model trained to distinguish cats from dogs learns the boundary between them, that is, the features that separate one class from the other. A generative model instead learns what cats look like and what dogs look like. The discriminative model can tell you which side of the line a new animal falls on; the generative model can tell you how unusual that animal is, even if it has never seen anything quite like it before.

This distinction has been explored in a range of literature, with one such paper of note being Ng. A. Y, Jordan, M. I (2001). Here the authors show that while discriminative models converge faster with labelled data, generative models have an inherent advantage in low-data regimes and in tasks that require understanding the structure of the input space — precisely the conditions that characterise cases like insider threat detection, rare and suspicious network connections, and user entity behaviour analytics (UEBA).

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

Jebara, T. (2004). Machine Learning: Discriminative and Generative. Kluwer Academic Publishers.

2.2 The Language Bottleneck

LLMs are generative models trained on the joint distribution of language tokens. A common architecture choice for this is the modern transformer, which is optimised for the compositional and sequential structure of natural language. [Vaswani A., 'Attention is all you need' (2017)] This is a remarkable engineering achievement for language tasks. However, it is a statistical liability when applied naively to all security telemetry. We can explore this with an example.

Consider a user with 50 failed login attempts within a five-minute window. A rule of this kind is a common security detector for possible brute force attacks. An LLM approach tokenises this data as text: "a user had fifty failed login attempts in five minutes." (Note that an LLM could also tokenise the raw event logs, taking into account aspects such as Windows Event ID 4625 and accompanying timestamps; the essence remains the same.) The numeral 50 (or 4625) becomes a token, just another word. The model must infer from linguistic context that "fifty" is anomalous, that it relates to a five-minute window, and that this combination is a threat signal. This multi-layer mapping from natural numerical data to tokenised language to security relevance is not only extremely inefficient but also fundamentally misaligned. By converting numerical quantities into linguistic tokens, values become words stripped of their quantitative grounding. This leaves a language model unable to compute rates, compare values against baselines, or adapt as environmental norms shift; operations that are trivial for purpose-built detectors but fundamentally opaque to a language model reasoning by static semantic association.

Let's consider an alternative approach which encapsulates our notion of "right tool for the right job". A purpose-built generative model such as a Hidden Markov Model trained on the authentication data can directly capture the temporal distribution of login attempts per unit time.

It directly calculates the likelihood of observing 50 attempts against a user's learned baseline. No tokenisation. No linguistic detour. Just direct statistical comparison against observed behaviour. This approach has many benefits.

1) It is immediately explainable. We can show you the distribution of normal numbers of failed logons, for that user, groups of similar users, for that time of day, we have full control over this.

2) We can directly quantify a meaningful and bounded "score". How rare is this observation, grounded in statistics.

3) It can adapt to changing baselines quickly. A host of new interns onboard and failed logons spike, we can directly quantify this evolution and adapt, without needing extensive token-usage to ground or explain away behaviour that can be seen in one plot.

4) For a given input, the model will ALWAYS produce the same output. No accounting of prompt phrasing, role allocation or amount of "don't make a mistake"s.

5) It is several orders of magnitude cheaper and easier to run computationally, owing to lower inference and training costs, no prompt overhead, and the elimination of tokens spent on "reasoning" alone. Purpose-built models also demand significantly less hardware, operating over a compact state space rather than needing billions of parameters.

These extensive benefits arise in this example use-case because we are selecting the appropriate method for what we need. No model will be best for all use-cases, including LLMs or any other foundation model.

> **Research finding:** *A 2025 benchmark study by Simbian found that top LLMs achieve 61–67% performance on real-world SOC investigation scenarios, leaving a significant gap for models better suited to the underlying data structure.*

Simbian (2025). AI in the SOC: Benchmarking LLMs for Autonomous Alert Triage. https://simbian.ai/blog/the-first-ai-soc-llm-benchmark

3. Applications of Generative Models in Security

The following four application areas represent domains where purpose-built generative models will very likely offer substantial benefits over LLMs. This is not because LLMs are generally inferior, but because the underlying statistical problem is quantitative, not linguistic. As an aside, there is increasing research into transformer architecture for tabular data. This is actually in-line with the message here, which is that the broad learning approach is powerful, but the substrate of language, words and characters, is the restrictive component [Hollmann N. et al. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second (2023), Qu J. et al. TabICLv2: A better, faster, scalable, and open tabular foundation model (2026)]. For the purpose of this paper, we will restrict our focus to more classical generative modelling approaches in the following examples.

1. Anomaly Detection with Interpretable Scores

A Gaussian Mixture Model trained on API call sequences returns log-likelihood scores — a quantitative, interpretable measure of how far an observation deviates from learned normal behaviour. When a SOC analyst asks “why is this alert firing?”, the model can point to which component of the learned distribution was violated. No black box. No language-model hallucination. Direct statistical accountability.

2. Temporal Behaviour Modelling

Hidden Markov Models capture P(state_t | state_{t-1}, observations) for session analysis. They learn the transition probabilities between behavioural states (login, file access, lateral movement) making them natural detectors of multi-step attack chains. When today’s behaviour diverges from the learned historical distribution, the model surfaces a drift signal with a computable probability score.

3. Behavioural Simulation for Red Teaming

Variational Autoencoders trained on network traffic can generate novel-but-plausible attack flows for testing defences — because they have learned the true statistical distribution of the traffic, not a linguistic approximation of it. This is generative modelling’s core strength: creating representative samples that preserve the statistical properties of the original data.

4. High-Dimensional Anomaly Detection

Autoencoders learn a compressed representation of normal behaviour. High reconstruction error signals anomaly. Naive Bayes classifiers learn P(features | class) and provide fast, sample-efficient classification with interpretable probability scores. These models operate directly in the feature space — numerical, categorical, temporal — without the linguistic detour that LLMs require.

4. A Field Guide to Security-Relevant Generative Models

The table below maps four primary examples of classical generative model architectures to their security use cases, distinguishing each by what probability distribution they learn and why that matters operationally.

Hidden Markov Models (HMMs). Learn P(observation_t | hidden_state_t) and P(state_t | state_{t-1}). Optimal for authentication sequences, session behaviour analysis, and multi-step attack chain detection. HMMs explicitly model temporal state transitions, the exact structure that characterises attacker dwell and lateral movement.

Naive Bayes. Learns P(features | class) under conditional independence assumptions. Extremely sample-efficient and interpretable. Optimal for fast classification of emails, URLs, and file hashes at scale. Provides calibrated probability scores that integrate cleanly with SIEM triage workflows.

Gaussian Mixture Models (GMMs). Learn P(X) as a weighted sum of Gaussian distributions, enabling soft clustering with explicit density estimation. Optimal for profiling user behaviour and

network traffic patterns. Unlike k-means, GMMs provide a probabilistic membership score that quantifies how anomalous a new observation is.

Variational Autoencoders (VAEs). Learn a latent distribution P(Z | X) and a generative distribution P(X | Z). Optimal for high-dimensional anomaly detection and synthetic data generation. When a VAE cannot accurately reconstruct an observation from its latent representation, that reconstruction error is a statistically grounded anomaly signal.

Critically, none of these models require tokenisation, language priors, or next-word prediction. Each learns the probability distribution directly relevant to the security task.

5. When LLMs are, and are not, the Right Tool

LLMs have genuine, valuable applications in the SOC. A comprehensive 2025 survey of LLM integration into SOC workflows found strong performance in log summarisation, report generation, threat intelligence extraction from news feeds, and analyst-facing Q&A interfaces. These are fundamentally language tasks, and LLMs are well-suited to them.

> *Nikitin et al. (2025). Large Language Models for Security Operations Centers: A Comprehensive Survey. arXiv:2509.10858.*

The challenge arises when LLMs are deployed for detection, the core real-time task of identifying whether a given sequence of events constitutes a threat. The evidence here is mixed at best. SentinelOne's 2026 analysis of LLM benchmarks for security found that models excelling at coding and mathematics provide minimal direct gains on security-specific tasks, and that general LLM capabilities do not readily translate to analyst-level threat reasoning.

> *SentinelOne Labs (2026). LLMs in the SOC (Part 1): Why Benchmarks Fail Security Operations Teams.*

> **Recommended split:** *Use LLMs for tasks where language is key: log summarisation, natural language threat intel, report drafting, analyst interfaces. Use purpose-built generative models for: real-time anomaly scoring, UEBA, session analysis, alert triage prioritisation.*

6. Conclusion: The Right Model for the Right Problem

The security industry's adoption of generative AI has been swift, but in many areas, superficial. The word "generative" has been co-opted to mean "LLM-based," obscuring a 200-year statistical tradition that contains exactly the tools the SOC needs.

The challenges facing modern security operations are not just a language problem. In many cases there is a problem of quantifying deviations from normal. Framed statistically, this is a

quantification of the gap between the probability distribution of normal user behaviour and the probability distribution of attacker behaviour. That gap is measured in authentication timestamps, API call sequences, event counts per minute, and lateral movement patterns. These are numerical, temporal, and stateful signals. They require models that learn numerical, temporal, and stateful distributions.

When a user deviates from their baseline, logging in at 3am, accessing systems they've never touched, transferring files at ten times their normal rate, the detection system must answer: "How improbable is this, given everything we know about this user's history?" That question has a precise statistical answer. Hidden Markov Models, Gaussian Mixture Models, and Variational Autoencoders were purpose-built to compute it. LLMs were not. Now, LLMs can be used for gaining insights and context from the linguistic components, but this is a careful implementation of "right tool for the right job".

So, this is not an argument against AI in the SOC. It is an argument for precision in AI model selection. The organisations that will lead in security operations over the next decade are not those that adopted the most prominent AI technology. They are those that matched model architecture to problems with rigour.

The cost of getting this wrong is quantifiable. Alert fatigue, driven by high false-positive rates from poorly calibrated detection models, costs the industry billions annually in analyst hours, analyst turnover, and breaches that dwelt undetected. The SANS 2024 SOC Survey found that 66% of SOC teams cannot keep pace with their alert volumes. The ACM's 2024 survey on alert fatigue in SOCs projects cybercrime damages reaching $13.82 trillion by 2028 — a number that scales with every detection model that forces security behaviour through a language bottleneck.

> *ACM Computing Surveys (2024). Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities.*

The good news is that the statistical tools to close this gap are mature, interpretable, and proven. They predate the LLM wave by decades. They do not hallucinate. They produce probability scores that an analyst can interrogate, audit, and trust. And they scale, because reducing 10,000 alerts to the 500 that genuinely warrant attention is not just a language task. It is a broad and complex inference task, and it calls for a range of generative models to perform it.

> **The closing principle:** *The right generative model is not the most famous one. It is the one whose core principles and assumptions most closely align with the structure of the problem you are trying to solve. For cybersecurity behavioural analytics, that model is likely not an LLM.*

References

Ng, A. Y., & Jordan, M. I. (2001). On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. Proceedings of NIPS '01. MIT Press.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

MIT News (2023). Explained: Generative AI. https://news.mit.edu/2023/explained-generative-ai-1109

Jalalvand, F., et al. (2024). Alert prioritisation in security operations centres: A systematic survey on criteria and methods. ACM Computing Surveys, 57(2), 1–36.

ACM Computing Surveys (2025). Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities. https://dl.acm.org/doi/10.1145/3723158

Nikitin et al. (2025). Large Language Models for Security Operations Centers: A Comprehensive Survey. arXiv:2509.10858.

Mdpi Journal of Cybersecurity and Privacy (2025). AI-Augmented SOC: A Survey of LLMs and Agents for Security Automation. https://doi.org/10.3390/jcp5040095

SANS Institute (2024). 2024 SOC Survey.

ISC2 (2024). Cybersecurity Workforce Study.

Simbian (2025). AI in the SOC: Benchmarking LLMs for Autonomous Alert Triage. https://simbian.ai/blog/the-first-ai-soc-llm-benchmark

SentinelOne Labs (2026). LLMs in the SOC (Part 1): Why Benchmarks Fail Security Operations Teams.

Baruwal Chhetri, M., et al. (2024). Towards human-AI teaming to mitigate alert fatigue in security operations centres. ACM Trans. Internet Technol., 24(3).

Hollmann, N., Müller, S., Eggensperger, K. and Hutter, F. (2023) 'TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second', The Eleventh International Conference on Learning Representations (ICLR 2023). Available at: https://arxiv.org/abs/2207.01848.

Qu, J., Holzmüller, D., Varoquaux, G. and Le Morvan, M. (2026) 'TabICLv2: A better, faster, scalable, and open tabular foundation model', arXiv preprint arXiv:2602.11139

Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin, 'Attention is All You Need', 2017, Available at: https://arxiv.org/pdf/1706.03762.pdf