

CyberSecLLM: A Hybrid Mamba–Transformer Foundation Model with ODE-Augmented State Dynamics for Zero-Shot Threat Intelligence

Roger Nick Anaedevha and Alexander Gennadevich Trofimov

Abstract—Contemporary intrusion detection systems rely on supervised paradigms that require extensive labeled data and rapidly become obsolete as adversaries evolve. This paper introduces CyberSecLLM, a family of domain-specific foundation models (3B, 7B, 13B parameters) pre-trained on six publicly available security datasets spanning cloud, industrial IoT, and enterprise environments. The architecture makes three contributions. First, it proposes ODE-augmented selective state space (ODE-S6) blocks, which embed neural ordinary differential equation solvers with temporal adaptive batch normalization within Mamba layers to model the continuous-time evolution of a latent threat state between discrete, irregularly timed flow observations, unifying the discrete selective scan with nonlinear continuous dynamics. Second, it interleaves these blocks with sparse cross-attention transformer layers grounded in a MITRE ATT&CK knowledge base and task-specialized mixture-of-experts routing. Third, the pre-training procedure combines five complementary objectives—masked flow modeling, contrastive attack discrimination, temporal forecasting, ATT&CK technique attribution, and a novel deep spatio-temporal point process loss that explicitly models inter-event attack timing via a neural conditional intensity function—balanced through learned uncertainty weights. Evaluation on CyberSecBench, an eleven-task benchmark incorporating three established LLM security suites (CTIBench, SecBench, CyberMetric), demonstrates that the 7B variant achieves 89.1% average zero-shot accuracy. Against specialized same-modality baselines on flow-level tasks—gradient-boosted trees, graph neural network intrusion detectors, and self-supervised traffic transformers—CyberSecLLM matches or exceeds supervised specialists while operating zero-shot across all tasks simultaneously; on the text-based benchmark subset, where large language models constitute a like-for-like comparison, it improves over GPT-4 by 18.6 points on average and over Foundation-Sec-8B by 12.4 points. Synthetic pre-training data are generated exclusively with open-weight models and validated with conditional-dependency and train-on-synthetic-test-on-real fidelity tests, ruling out distillation contamination of the evaluation. The architecture processes million-token network traces in 2.3 seconds on a single A100, a 19× speedup over quadratic transformers. All code, models, and the evaluation suite are released openly to support reproducibility.

Keywords— *Foundation models, cybersecurity, intrusion detection, state space models, neural ordinary differential equations, temporal point processes, mixture-of-experts, zero-shot learning, network traffic analysis.*

I. INTRODUCTION

The cybersecurity landscape is defined by fundamental asymmetry: defenders must protect against all possible attack vectors, whereas adversaries need exploit only a single vulnerability. Modern security operations centers (SOCs) process millions of network flow records daily, yet conventional intrusion detection systems remain anchored to supervised classification on static labeled datasets that cannot anticipate novel threats. The gap between the rate at which adversarial techniques evolve—the MITRE ATT&CK framework now catalogues over 750 techniques and sub-techniques—and the rate at which detection models are retrained creates a persistent window of exposure.

Foundation models offer a paradigm shift: by pre-training on large, diverse corpora to learn transferable representations, they can generalize to unseen tasks without task-specific fine-tuning. Yet applying foundation models to network security faces distinct challenges. Network traffic is observed as a *marked temporal point process*: a sequence of discrete flow events stamped with continuous-valued, irregularly spaced arrival times. Although no packet exists between arrivals, the temporal spacing between events carries causal information about attack progression, and the latent state of the monitored system—attacker position, campaign phase, propagation of a botnet—continues to evolve between observations. A brute-force campaign produces bursts of authentication events at irregular intervals; a lateral-movement phase may span hours of sparse activity. Discrete-time models that tokenize flows into fixed windows discard this information. In our prior work we introduced continuous-time temporal graph neural networks[1] and investigated Neural ODE architectures for dynamical-system modeling[2], demonstrating that continuous-time representations substantially improve temporal pattern recognition in evolving graph structures. CyberSecLLM extends these foundations to the scale of modern foundation models by integrating continuous-time ODE dynamics directly within selective state-space layers.

Specifically, this paper makes four contributions. *First*, we propose ODE-augmented selective state space (ODE-S6) blocks that embed adaptive-step Neural ODE solvers with temporal

adaptive batch normalization (TA-BN)[3] within Mamba’s selective scan mechanism[4]. This unifies the continuous-time state equation underlying SSMs with nonlinear dynamics, capturing irregular temporal patterns in network traffic while maintaining Mamba’s linear-time complexity. *Second*, we introduce a five-objective pre-training framework extending standard masked and contrastive objectives with a *deep spatio-temporal point process* (DSTPP) loss. Drawing on the Neural Hawkes Process[5] and Transformer Hawkes Process[6], the DSTPP loss models the conditional intensity of security events to capture self-exciting attack contagion and temporal clustering, providing a principled probabilistic objective for learning inter-event timing representations. *Third*, we construct CyberSecBench, an eleven-task evaluation suite that integrates our eight original detection and triage tasks with three established LLM security benchmarks—CTIBench[7], SecBench[8], and CyberMetric[9]—enabling rigorous comparison against both general-purpose and security-specific models on over 63,000 evaluation items. *Fourth*, we conduct a modality-stratified evaluation: on flow-level tasks (T1–T8) the 7B-parameter CyberSecLLM is compared against specialized same-modality baselines—gradient-boosted trees, GNN-based intrusion detectors, and self-supervised traffic transformers—which it matches or exceeds while remaining zero-shot and multi-task; on text-based benchmarks (T9–T11), where large language models are directly comparable, it improves over GPT-4 by 18.6 points on average and over Foundation-Sec-8B by 12.4 points, while processing million-token sequences in 2.3 seconds. Averaged over all eleven tasks it attains 89.1%.

The remainder of this paper is organized as follows. Section II surveys related work and identifies the gaps that motivate our design. Section III formalizes notation and the problem statement. Section IV details the proposed architecture. Section V describes datasets and the evaluation benchmark. Section VI presents experimental results. Section VII discusses implications, and Section VIII concludes.

II. RELATED WORK

We survey four research threads whose intersection defines the contribution space of CyberSecLLM, consolidating eight distinct lines of prior work.

A. Domain-specific LLMs and deep learning for network traffic

Early efforts to apply language models to security repurposed general-purpose architectures. SecureBERT[10] continued pre-training on cybersecurity text corpora, achieving strong NER results but lacking network-level reasoning. SecGPT[11] and SecurityLLM[12] fine-tuned GPT-family models on curated CTI datasets. Foundation-Sec-8B[13] introduced security-specific pre-training at scale but employed a standard transformer backbone, inheriting quadratic attention complexity and struggling with long network traces. For raw traffic, ET-BERT[29] and NetGPT[30] applied transformer pre-training to encrypted-traffic classification and traffic generation; TrafficLLM[31] introduced generic traffic representations for LLMs; and ContinuumGNN[32] applied spatio-temporal graph

autoencoders to APT detection. Critically, *no prior security LLM operates on raw network flow representations at the pre-training level*, and none combines continuous-time dynamics, sparse expert routing, and multi-task pre-training on raw flow features.

B. State-space models and Mamba

Structured state-space models (S4)[14] demonstrated that the linear state-space ODE with HiPPO initialization[37] captures long-range dependencies in $O(L)$ time via parallel scan or FFT convolution, overcoming the quadratic cost of self-attention. Mamba[4] introduced input-dependent selectivity, making **B, C, Δ** functions of the current token and breaking the time-invariance assumption; Mamba-2[15] revealed a structured duality between SSMs and attention, enabling chunkwise computation with expanded state dimensions. Liquid structural SSMs[44] further generalized the recurrence to time-varying continuous dynamics. However, all of these discretize via zero-order hold (ZOH), which assumes *piecewise-constant* input between time steps—an assumption violated by irregularly spaced network events where inter-arrival times vary across eight orders of magnitude. Our ODE-S6 block addresses this by solving the continuous dynamics directly with adaptive-step integration, generalizing the SSM formulation to the irregularly-sampled regime.

C. Neural ODEs, hybrid architectures, and mixture of experts

The Neural ODE framework[16] and its extensions—latent SDEs[17], augmented ODEs[18], controlled ODEs for irregular time series[19]—established the adjoint-based training paradigm. A persistent difficulty, the failure of standard batch normalization under variable solver grids, was resolved by Zheng et al.[3] through TA-BN, which interpolates normalization statistics across fixed anchor time points. Anomaly detection in continuous-time temporal provenance graphs[45] demonstrated that continuous-time dynamics materially outperform discrete baselines on security-relevant event streams. Hybrid architectures—Jamba[24], Zamba[25], MoE-Mamba[26]—interleave Mamba and attention layers, finding that the combination outperforms either alone; DeepSeekMoE[28] and Switch Transformers[27] demonstrated sparse expert routing that decouples total capacity from per-token computation. CyberSecLLM is the first to embed Neural ODE solvers with TA-BN normalization within selective SSM blocks at foundation-model scale, combining these strands into a unified hybrid architecture with security-task-specialized routing.

D. Temporal point processes and LLM security benchmarks

Hawkes processes have been applied to cyber event modeling by Baldwin et al.[20], who found strong evidence of self-exciting contagion across network ports; Peng et al.[21], who modeled extreme attack rates via marked point processes; and Ferraro et al.[22], who applied neural TPPs to industrial-control-system attack prediction. Neural extensions—the Neural Hawkes Process[5], Self-Attentive Hawkes Process[23], and Transformer Hawkes Process[6]—offer learnable intensity

functions with inhibition and long-range memory. *However, no foundation model has incorporated a point process pre-training objective to explicitly learn inter-event timing patterns.*

The LLM security evaluation landscape has matured with CTIBench[7] (5,610 CTI tasks across five categories), SecBench[8] (47,910 multi-dimensional questions), and CyberMetric[9] (10,000 expert-validated MCQs). Meta’s CyberSOCEval[33] assesses SOC-relevant malware analysis, and CAIBench[34] reveals a gap between knowledge scores (70–89%) and practical task completion (31%)—motivating evaluations that bridge both dimensions. For data augmentation, SynthCTI[35] demonstrated that LLM-generated CTI text improves ATT&CK classification by 48.6%, while Knowledge-to-Data[36] conditioned LLMs on protocol documentation to synthesize structured traffic without fine-tuning. CyberSecBench integrates these streams with our eight original flow-level tasks to enable rigorous comparison across both general-purpose and security-specific models.

III. PRELIMINARIES AND PROBLEM FORMULATION

Building on the gaps identified above, this section establishes the mathematical foundations, notation conventions, and formal problem statement upon which the proposed architecture rests.

A. Network flow sequences as continuous-time event streams

Let $\mathcal{S} = \{(f_i, t_i, y_i)\}_{i=1}^N$ denote a network flow corpus, where each flow $f_i \in \mathbb{R}^D$ is a D -dimensional feature vector (packet statistics, protocol flags, port numbers, behavioral indicators), $t_i \in \mathbb{R}^+$ is the continuous-valued timestamp, and $y_i \in \mathcal{Y} = \{0, 1, \dots, K\}$ is an optional attack category label ($y_i = 0$ for benign traffic, $y_i = k$ for attack type k from a unified taxonomy $\mathcal{Y}$ of K classes). Each flow decomposes into four semantic components:

$$f_i = (f_i^{\text{cont}}, f_i^{\text{cat}}, t_i, f_i^{\text{topo}}) \quad (1)$$

where $f_i^{\text{cont}} \in \mathbb{R}^{D_c}$ denotes continuous features (packet counts, byte volumes, flow duration), $f_i^{\text{cat}} \in \mathbb{N}^{D_q}$ denotes categorical features (protocol, source/destination port, TCP flags), $t_i \in \mathbb{R}^+$ is the timestamp, and $f_i^{\text{topo}} \in \{0, 1\}^{|\mathcal{V}|}$ encodes the topological neighborhood in the communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

We emphasize a modeling distinction that underpins the architecture. The *observations* $\{(f_i, t_i)\}$ are discrete: no flow record exists between arrivals, and a monitoring system indeed waits for the next event. What evolves in continuous time is not the data stream but the *latent state* of the monitored environment—the progression of an attack campaign, the propagation of malware, the decaying relevance of stale observations. Modeling a continuously evolving latent state observed only at discrete, irregularly spaced instants is a well-established paradigm: ODE-RNNs and latent ODEs[47], GRU-ODE-Bayes[48], and neural controlled differential equations[49] evolve a hidden state between observations by numerical integration, while the neural Hawkes process[5] lets its continuous-time LSTM state decay analytically between events, and neural jump stochastic differential equations[50] formalize hybrid systems whose latent trajectory flows

continuously and jumps at event arrivals. Because inter-arrival intervals $\delta_i = t_i - t_{i-1}$ encode discriminative structure—beaconing periodicity, burstiness, low-and-slow campaigns—discrete-index models that see only event order discard this signal, whereas the continuous-time formulation consumes it natively.

B. Continuous-time state-space models

A linear time-invariant SSM maps a scalar input $x(t) \in \mathbb{R}$ to output $y(t) \in \mathbb{R}$ through a latent state $h(t) \in \mathbb{R}^{N_s}$:

$$dh(t)/dt = \mathbf{A} h(t) + \mathbf{B} x(t), \quad (2)$$

$$y(t) = \mathbf{C} h(t) + \mathbf{D} x(t), \quad (3)$$

where $\mathbf{A} \in \mathbb{R}^{N_s \times N_s}$ is initialized via HiPPO[37], $\mathbf{B} \in \mathbb{R}^{N_s \times 1}$, $\mathbf{C} \in \mathbb{R}^{1 \times N_s}$, and $\mathbf{D} \in \mathbb{R}$ is a feedthrough (typically omitted). Equation (2) is discretized using step $\Delta > 0$ via zero-order hold:

$$\bar{\mathbf{A}} = \exp(\Delta \mathbf{A}), \quad (4)$$

$$\bar{\mathbf{B}} = (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B}, \quad (5)$$

yielding $h_k = \bar{\mathbf{A}} h_{k-1} + \bar{\mathbf{B}} x_k$, $y_k = \mathbf{C} h_k$. Mamba’s selectivity[4] makes $\mathbf{B}, \mathbf{C}, \Delta$ functions of the input:

$$\mathbf{B}_k = s_B(x_k), \quad \mathbf{C}_k = s_C(x_k), \quad \Delta_k = \text{softplus}(\tau + s_\Delta(x_k)), \quad (6)$$

where s_B, s_C, s_Δ are linear maps and τ is a learned bias. Input dependence breaks time-invariance, producing a time-varying system that selectively attends to or ignores tokens.

C. Neural ordinary differential equations

A Neural ODE[16] parameterizes continuous-time dynamics as

$$dh(t)/dt = g_\theta(h(t), t), \quad h(t_0) = h_0, \quad (7)$$

with $g_\theta: \mathbb{R}^{d_h} \times \mathbb{R} \rightarrow \mathbb{R}^{d_h}$ a neural network. The state at $t_1 > t_0$ is $h(t_1) = h(t_0) + \int_{t_0}^{t_1} g_\theta(h(\tau), \tau) d\tau$, computed by adaptive solvers such as Dormand–Prince (dopri5). Gradients use the adjoint method: $a(t) = \partial \mathcal{L} / \partial h(t)$ with $da/dt = -a(t)^\top (\partial g_\theta / \partial h)$, achieving $\mathcal{O}(1)$ memory by solving the augmented ODE backward in time.

Temporal adaptive batch normalization (TA-BN)[3] addresses the mismatch between batch normalization’s fixed-evaluation-point assumption and adaptive solvers’ variable time grids. Our implementation parameterizes the affine parameters as continuous functions of time via MLPs with periodic features. With $\psi(t) = [t, \sin(\omega t), \cos(\omega t)]^\top$ (ω learnable), time-dependent affine parameters are $\gamma(t) = \text{softmax}(\text{MLP}_\gamma(\psi(t)))$ and $\alpha(t) = \text{MLP}_\alpha(\psi(t))$. Normalization at solver time t_j :

$$\begin{aligned} & \text{TABN}(x; t_j) \\ &= \gamma(t_j) \cdot x - \mu(t_j) / \sqrt{\sigma^2(t_j) + \epsilon} + \alpha(t_j), \end{aligned} \quad (8)$$

where μ, σ^2 are mini-batch statistics during training (with EMA updates) and running population statistics at inference. The softmax on γ bounds scales; periodic features capture cyclic traffic patterns.

D. Temporal point processes

A temporal point process models a sequence of discrete events at continuous timestamps. Given history $\mathcal{H}(t) = \{(t_i, k_i) : t_i < t\}$ with types $k_i \in \{1, \dots, K\}$, the conditional

intensity $\lambda_k^*(t)$ (with $*$ denoting history-conditioning) specifies the instantaneous expected rate of type- k events:

$$\lambda_k^*(t) = \lim_{\delta \rightarrow 0} \frac{\Pr(\text{type-}k \text{ event in } [t, t + \delta) \mid \mathcal{H}(t))}{\delta}. \quad (9)$$

The log-likelihood of $\{(t_i, k_i)\}_{i=1}^n$ over $[0, T]$ is

$$\mathcal{L}_{\text{TPP}} = \sum_{i=1}^n \log \lambda_{k_i}^*(t_i) - \sum_{k=1}^K \int_0^T \lambda_k^*(\tau) d\tau, \quad (10)$$

rewarding high intensity at observed events and penalizing it where none occur. The classical Hawkes process[38] models self-excitation via triggering kernels, capturing the contagion dynamics empirically observed in cyber attack sequences[20].

E. Problem statement

Given: (i) a pre-training corpus $\mathcal{D}_{\text{pre}} = \bigcup_{j=1}^6 \mathcal{D}_j$ comprising six network-security datasets with $N > 86 \times 10^6$ flow records, harmonized to a common D -dimensional feature schema ($D = 80$); (ii) an external threat-intelligence knowledge base $\mathcal{K} = \{\kappa_m\}_{m=1}^{M_K}$ of pre-computed embeddings representing MITRE ATT&CK techniques and CVE descriptions; (iii) a set of target tasks $\mathcal{T} = \{\tau_1, \dots, \tau_{11}\}$ spanning intrusion detection (multi-class, cross-dataset, OOD), threat intelligence (CVE severity, ATT&CK attribution, malware family), operational security (alert triage, incident summarization), and external LLM benchmarks (CTIBench, SecBench, CyberMetric).

Objective: Learn a foundation model $\Phi_\theta: \mathcal{X} \rightarrow \mathcal{Y}$ that, without task-specific fine-tuning, maximizes aggregate zero-shot performance across all $|\mathcal{T}|$ tasks:

$$\begin{aligned} \theta^* = \operatorname{argmin}_{\theta} \quad & \mathcal{L}_{\text{pre}}(\theta; \mathcal{D}_{\text{pre}}, \mathcal{K}) \\ \text{s.t.} \quad & 1/|\mathcal{T}| \sum_{\tau \in \mathcal{T}} \text{Perf}(\Phi_\theta, \tau) \rightarrow \max, \end{aligned} \quad (11)$$

where $\mathcal{L}_{\text{pre}}$ is the combined pre-training loss (Section IV-E) and $\text{Perf}(\cdot, \tau)$ is the task-specific metric (accuracy, F1, AUROC, nDCG, or ROUGE-L).

Constraints: The model must process sequences of length $L \leq 10^6$ tokens in sub-quadratic time ($\mathcal{O}(L)$) and fit within single-accelerator memory (≤ 80 GB) at the 7B parameter scale.

F. Theoretical properties of the ODE-S6 block

Two properties, proved in the Supplementary Material, underpin the soundness of the architecture.

Let the ODE-S6 dynamics be defined by Eq. (14) with TA-BN parameters bounded by the stability regularizer (18). Then for any two initial states $h_0, h_0' \in \mathbb{R}^{d_h}$ and any integration horizon $\tau \leq T$, the induced flow Φ_τ satisfies

$$\|\Phi_\tau(h_0) - \Phi_\tau(h_0')\| \leq e^{L_g \tau} \cdot \|h_0 - h_0'\|, \quad (12)$$

where $L_g = \sum_{s=1}^S \alpha_s L_{\phi_s}$ is the effective Lipschitz constant of the multi-scale dynamics and L_{ϕ_s} is bounded in closed form by the Frobenius norms of the TA-BN MLP weights. Consequently, the adjoint state $a(t) = \partial \mathcal{L} / \partial h(t)$ remains bounded along the backward pass, and gradients do not diverge.

For a sequence of length L with S time scales and an adaptive solver requiring N_{fe} function evaluations per inter-event interval, the forward pass of a single ODE-S6 block has time

complexity $\mathcal{O}(L \cdot N_{\text{fe}} \cdot S \cdot d_h)$ and memory complexity $\mathcal{O}(d_h)$ under adjoint backpropagation. Since N_{fe} and S are bounded constants independent of L , this yields *linear* scaling in sequence length, matching the asymptotic efficiency of the base Mamba selective scan.

Theorem 1 is a direct consequence of Grönwall's inequality applied to the bounded-Lipschitz multi-scale vector field; Theorem 2 follows from the adjoint-method analysis of [16]. Together they guarantee that the proposed block inherits the training stability and computational efficiency of standard Mamba while extending it to the irregularly-sampled continuous-time regime.

IV. PROPOSED METHODOLOGY

Figure 1 illustrates the CyberSecLLM architecture, consisting of four components: multi-pathway flow tokenization (Section IV-A), ODE-augmented selective state space blocks (Section IV-B), knowledge-grounded cross-attention with MoE feedforward (Sections IV-C–IV-D), and five-objective pre-training (Section IV-E).

A. Multi-pathway flow tokenization

Each flow f_i is encoded through four parallel pathways that respect the heterogeneous nature of network features.

Continuous pathway. A two-layer GELU MLP maps the D_c -dimensional continuous features to $d/4$ dimensions: $z_i^{\text{cont}} = W_2 \cdot \text{GELU}(W_1 f_i^{\text{cont}} + b_1) + b_2$ with $W_1 \in \mathbb{R}^{d_h \times D_c}$, $W_2 \in \mathbb{R}^{d/4 \times d_h}$.

Categorical pathway. Each categorical feature (protocol, source/destination port, TCP flags) is mapped through a learned embedding $E_q \in \mathbb{R}^{V_q \times d_e}$, concatenated, and projected: $z_i^{\text{cat}} = W_q \cdot \text{Cat}(E_{\text{proto}}[p_i], E_{\text{sport}}[s_i], E_{\text{dport}}[d_i], E_{\text{flag}}[g_i]) + b_q$.

Temporal pathway. Following [23], the timestamp is encoded via learnable sinusoidal features: $z_i^{\text{temp}} = W_t [\sin(\omega_1 t_i + \phi_1), \cos(\omega_1 t_i + \phi_1), \dots]^T + b_t$, frequencies $\{\omega_j\}$ and phases $\{\phi_j\}$ learnable.

Topological pathway. For flows involving host v_i , a single-hop mean aggregation over neighbors $\mathcal{N}(v_i)$ in the communication graph produces $z_i^{\text{topo}} = W_g \cdot |\mathcal{N}(v_i)|^{-1} \sum_{v_j \in \mathcal{N}(v_i)} e_{v_j} + b_g$.

The four encodings are concatenated and projected:

$$= \text{RMSNorm}(W_{\text{proj}} [z_i^{\text{cont}}, z_i^{\text{cat}}, z_i^{\text{temp}}, z_i^{\text{topo}}] + b_{\text{proj}}), \quad (13)$$

producing the token embedding $x_i \in \mathbb{R}^d$ that serves as input to the backbone.

B. ODE-augmented selective state-space blocks

The central architectural contribution is the ODE-S6 block, which replaces Mamba's fixed ZOH discretization with an adaptive Neural ODE solver featuring multi-scale temporal dynamics, enabling the model to capture irregular temporal patterns across eight orders of magnitude.

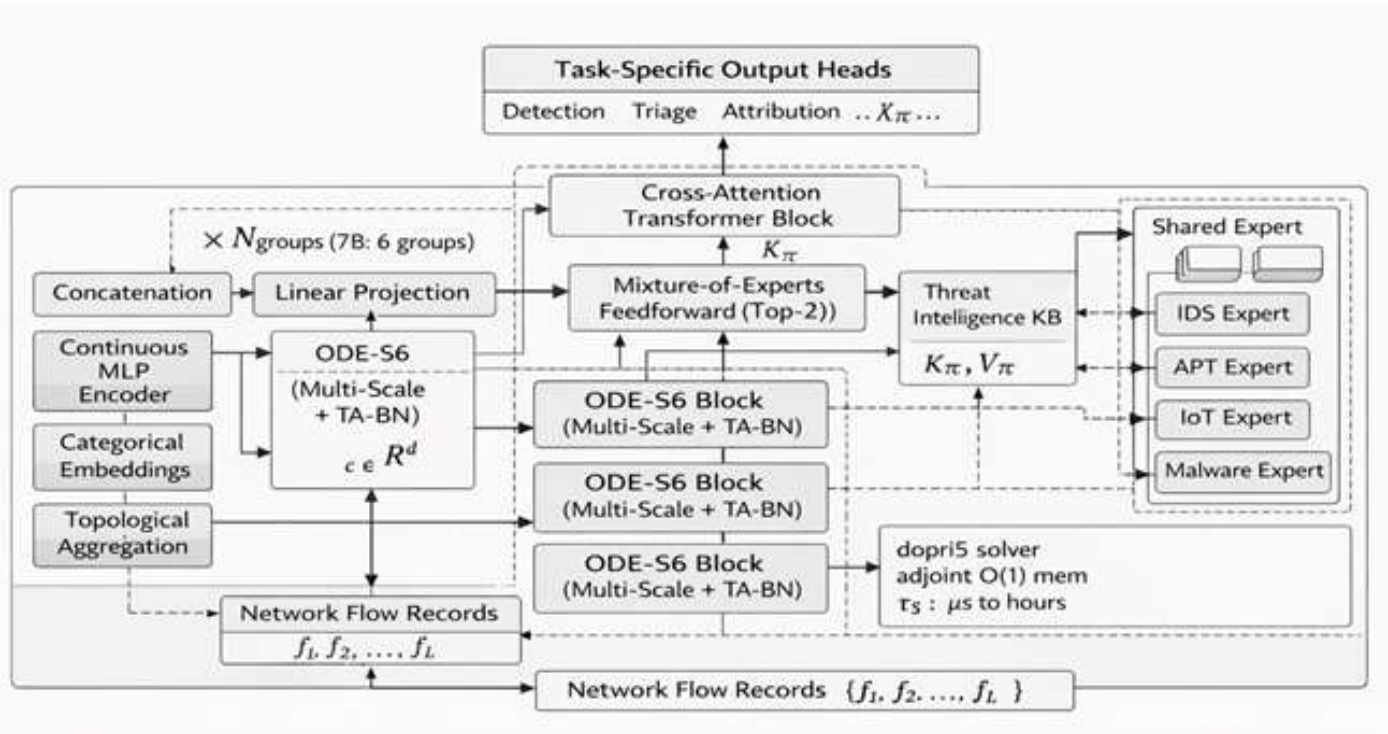


Fig. 1. Architecture of CyberSecLLM. Network flows are tokenized through four parallel pathways (bottom), processed through repeating groups of three ODE-S6 blocks, a cross-attention block grounded in the threat-intelligence KB $\mathcal{K}$, and a MoE feedforward layer with domain-specialized experts. Task heads (top) include a point-process head for inter-event timing $\lambda^*(t)$.

Motivation. The standard Mamba selective scan assumes approximately uniform time spacing through scalar Δ_k (6). In network traffic, inter-event intervals vary from microseconds during a DDoS burst to hours during reconnaissance. Fixed discretization either over-resolves quiet periods or under-resolves bursts. The ODE-S6 block therefore advances the latent state h by exactly the observed inter-arrival interval $\delta_k = t_k - t_{k-1}$, in the spirit of ODE-RNN and latent-ODE models for irregularly sampled series[47, 49] and of the analytically decaying hidden state of the neural Hawkes process[5]: the observations remain discrete events, while the model’s belief about the monitored system evolves continuously between them.

Multi-scale ODE dynamics. Given tokens $\{x_k\}_{k=1}^L$ with timestamps $\{t_k\}$, the ODE-S6 block defines continuous state evolution between consecutive events via multi-scale dynamics:

$$dh/d\tau = \sum_{s=1}^S \alpha_s \cdot g_{\phi_s}(\text{TABN}_s(h; \tau/\tau_s), \tau/\tau_s), \quad (14)$$

where $\{g_{\phi_s}\}$ are S parallel TA-BN-ODE branches at time scales $\{\tau_s\} = [10^{-6}, 10^{-3}, 1, 3600]$ s (spanning microsecond-level packet timing through hourly session patterns), $\alpha_s = \text{softmax}(\mathbf{a})_s$ are learned weights with $\mathbf{a} \in \mathbb{R}^S$, and each TABN_s applies temporal adaptive batch normalization at rescaled time. Each branch is a two-layer network with ELU activation:

$$g_{\phi_s}(h, t) = \text{ELU}(W_{2s} \cdot \text{TABN}_{2s}(\text{ELU}(W_{1s} \cdot \text{TABN}_{1s}(h; t)); t)). \quad (15)$$

The state at event time t_k is obtained by numerical integration from t_{k-1} :

$$h(t_k^-) = h(t_{k-1}) + \int_{t_{k-1}}^{t_k} \sum_{s=1}^S \alpha_s \cdot g_{\phi_s}(\text{TABN}_s(h(\tau); \tau/\tau_s), \tau/\tau_s) d\tau, \quad (16)$$

computed by Dormand–Prince (dopri5, $\text{rtol}=10^{-3}$, $\text{atol}=10^{-4}$) with adjoint backpropagation for $O(1)$ memory.

Event-driven discrete updates. After continuous integration yields $h(t_k^-)$, a discrete update incorporates the new event’s features:

$$h(t_k) = h(t_k^-) + \Delta h_k, \quad \Delta h_k = \text{SiLU}(W_u^{(2)} \text{SiLU}(W_u^{(1)} x_k + b_u^{(1)}) + b_u^{(2)}), \quad (17)$$

with $W_u^{(1)} \in \mathbb{R}^{d_h \times d}$, $W_u^{(2)} \in \mathbb{R}^{d_h \times d_h}$. This flow-and-jump formulation is the state-space analogue of neural jump stochastic differential equations[50]: the *latent* trajectory evolves continuously between events (capturing attacker progression and the decaying evidential weight of stale observations), while each discrete flow arrival triggers an instantaneous state update. We stress that this is a claim about the model’s latent belief state, not about the packet stream itself, which is discrete; continuous-time latent dynamics between discrete observations is precisely the regime for which ODE-RNNs[47], GRU-ODE-Bayes[48], and the neural Hawkes process[5] were developed, and for which point-process models of cyber events provide security-specific precedent[51, 52].

Stability regularization. To ensure gradient stability through the adjoint backward pass, we add L_2 regularization on the TA-BN MLP parameters:

$$\mathcal{L}_{\text{stab}} = \eta \sum_{s,l} (\|W_{\gamma,s,l}\|_F^2 + \|W_{\alpha,s,l}\|_F^2), \quad (18)$$

with $\eta = 10^{-4}$. This bounds the Lipschitz constant of the dynamics (Supplementary, Proposition 2).

Output. The output at position k combines the evolved state with an input-dependent projection:

$$y_k = \mathbf{C}_k^\top h(t_k) + \mathbf{D} \cdot x_k, \quad \mathbf{C}_k = s_c(x_k), \quad (19)$$

with $s_c: \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$ a linear projection.

Complexity. Each ODE solve requires $O(N_{fe} \cdot S)$ function evaluations per inter-event interval (typically $N_{fe} = 6-12$ for dopri5, $S = 4$). Total cost is $O(L \cdot N_{fe} \cdot S \cdot d_h)$, linear in sequence length; memory is $O(d_h)$ per step via the adjoint method.

C. Cross-attention to threat intelligence

After every three ODE-S6 blocks, a cross-attention transformer block grounds flow representations in the external knowledge base $\mathcal{K}$. Multi-head self-attention is followed by multi-head cross-attention where queries come from the flow representations $H \in \mathbb{R}^{L \times d}$ and keys/values from the KB embeddings $\mathcal{K} \in \mathbb{R}^{M_K \times d}$:

$$H^{\text{self}} = H + \text{MHA}_{\text{self}}(\text{LN}(H)), \quad (20)$$

$$H^{\text{cross}} = H^{\text{self}} + \alpha_{\text{gate}} \cdot \text{MHA}_{\text{cross}}(\text{LN}(H^{\text{self}}), \mathcal{K}), \quad (21)$$

with $\alpha_{\text{gate}} \in [0, 1]$ a learned scalar gate (initialized at 0.5) that controls external knowledge influence. When the KB is unavailable, the gate approaches zero and the block degrades gracefully to pure self-attention.

D. Task-specialized mixture-of-experts

Following each cross-attention block, a mixture-of-experts feedforward layer provides capacity scaling with sparse activation. Following DeepSeekMoE[28], the layer comprises E routed experts $\{e_j\}$ and E_s shared experts $\{e_j^s\}$ that are always active:

$$\begin{aligned} \text{MoE}(x) &= \sum_{j=1}^{E_s} e_j^s(x) + \sum_{j \in \text{top-K}(\pi(x), K_r)} \pi_j(x) \cdot e_j(x), \quad (22) \end{aligned}$$

where $\pi(x) = \text{softmax}(W_r x + b_r) \in \mathbb{R}^E$ is the router distribution, top-K selects K_r experts with highest routing probability, and each expert is a SwiGLU network[39]. An auxiliary load-balancing loss $\mathcal{L}_{\text{bal}} = E \cdot \sum_j p_j q_j$ prevents expert collapse (p_j : fraction of tokens to expert j ; q_j : average routing probability).

E. Multi-task pre-training with point-process objective

The pre-training procedure optimizes five complementary objectives simultaneously.

(O1) Masked Flow Modeling (MFM). 15% of tokens are replaced with a learnable [MASK]; the model reconstructs original features via MSE: $\mathcal{L}_{\text{MFM}} = |\mathcal{M}|^{-1} \sum_{k \in \mathcal{M}} \|f_k - \hat{f}_k\|_2^2$.

(O2) Contrastive Attack Discrimination (CAD). An InfoNCE loss clusters representations of same-type flows: $\mathcal{L}_{\text{CAD}} = -|\mathcal{P}|^{-1} \sum_{(i,j) \in \mathcal{P}} \log \frac{\exp(\text{sim}(z_i, z_j)/\tau_c)}{\sum_{k=1}^B \exp(\text{sim}(z_i, z_k)/\tau_c)}$, with $z_i = \text{MLP}(h_i^{\text{[CLS]}})$, cosine similarity, and $\tau_c = 0.07$.

(O3) Temporal Forecasting (TF). Given the first $L - k$ tokens, predict the next k : $\mathcal{L}_{\text{TF}} = k^{-1} \sum_{j=L-k+1}^L \|f_j - \hat{f}_j\|_2^2$.

(O4) ATT&CK Technique Attribution (AA). Multi-label BCE over 193 ATT&CK indicators: $\mathcal{L}_{\text{AA}} = -193^{-1} \sum_{m=1}^{193} [y_m \log \hat{y}_m + (1 - y_m) \log(1 - \hat{y}_m)]$.

(O5) Deep Spatio-Temporal Point Process (DSTPP). This novel objective models inter-event attack timing through a three-component neural conditional intensity function. A dedicated transformer encoder $\mathcal{F}_{\text{TTP}}$ (4 layers, 8 heads) captures long-range temporal dependencies. The conditional intensity for attack type k at time t decomposes additively into

$$\lambda_k^*(t) = \text{softplus}(\mu_k) + \text{softplus}(W_k^\top \mathcal{F}_{\text{TTP}}(h))$$

$$+ \sum_{i: t_i < t}^{\text{background}} \hat{\alpha}_{k_i, k} e^{-\hat{\beta}_{k_i, k}(t-t_i)}, \quad (23)$$

Hawkes excitation

where μ_k is a learnable background rate, W_k the neural intensity projection, $\hat{\alpha}_{k_i, k} = \text{softplus}(\alpha_{k_i, k})$ and $\hat{\beta}_{k_i, k} = \text{softplus}(\beta_{k_i, k})$ positive excitation and decay parameters, and k_i the mark of the i -th event. The Hawkes component captures cross-excitation—a brute-force attack can elevate lateral-movement intensity—mirroring the contagion dynamics of[20].

Log-barrier survival integral. The compensator $\Lambda^* = \sum_k \int_0^T \lambda_k^*(\tau) d\tau$ is approximated via equispaced quadrature with a log-barrier penalty:

$$\begin{aligned} \Lambda^* &\approx T/Q \sum_{q=1}^Q \sum_{k=1}^K \lambda_k^*(\tau_q), \quad \mathcal{L}_{\text{barrier}} \\ &= -\mu_b \sum_{q,k} \log \lambda_k^*(\tau_q), \quad (24) \end{aligned}$$

with $Q = 128$ quadrature points and $\mu_b = 0.01$. By Lemma 1 (Supplementary), error satisfies $|\Lambda^* - \hat{\Lambda}^*| \leq \lambda_\lambda T^2/(2Q)$, achieving $O(n \cdot Q) = O(n^{3/2})$ total cost vs. $O(n^2)$ pairwise. The complete DSTPP loss:

$$\mathcal{L}_{\text{DSTPP}} = -\sum_{i=1}^n \log \lambda_{k_i}^*(t_i) + \hat{\Lambda}^* + \mathcal{L}_{\text{barrier}}. \quad (25)$$

Combined loss. The five objectives plus stability regularization are balanced using learned homoscedastic uncertainty weights[40]:

$$\mathcal{L}_{\text{pre}} = \sum_{o=1}^5 1/2 \sigma_o^2 \mathcal{L}_o + \log \sigma_o + \beta \mathcal{L}_{\text{bal}} + \mathcal{L}_{\text{stab}}, \quad (26)$$

with σ_o learnable per-objective scalars, $\beta = 0.01$ weighting MoE load-balancing, and $\mathcal{L}_{\text{stab}}$ ($\eta = 10^{-4}$) the TA-BN stability regularization. Log-variance terms prevent degenerate solutions where any single objective dominates.

F. Training procedure

Pre-training uses AdamW[41] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1, gradient clipping at norm 1.0. The learning rate follows a cosine schedule with linear warmup (1% of total steps) from 0 to 3×10^{-4} and decay to 3×10^{-5} . FP16 mixed-precision training with dynamic loss scaling. Distributed

training uses DeepSpeed ZeRO-3[42] across 64 NVIDIA A100 80 GB GPUs with activation checkpointing. The ODE solver is dopri5 with relative tolerance 10^{-3} , absolute tolerance 10^{-4} , and max 20 function evaluations per inter-event interval. The 7B model trains for 750K steps (~ 4 weeks) on the full 86M-record corpus with an effective batch yielding ~ 8 M tokens per step via gradient accumulation.

V. DATASETS AND EVALUATION BENCHMARK

A. Pre-training corpus

The pre-training corpus comprises six publicly available network security datasets, harmonized to a common 80-feature schema via dataset-specific preprocessing pipelines. Table I summarizes each dataset.

TABLE I
Pre-training datasets.

Dataset	Records	Feat.	Domain	Attacks
CSE-CIC-IDS2018	16.2M	80	Enterprise	7 types
CIC-IoT-2023	46.7M	46	IoT	33 types
UNSW-NB15	2.5M	49	General	9 types
Edge-IIoTset	20.8M	61	Industrial	14 types
Azure Cloud	varies	40+	Cloud	APT
K8s/Docker	varies	35+	Container	Misc.
Total	86M+	80*		14 unified

*After harmonization to common schema. DOIs: IIS3D 10.34740/kaggle/dsv/12479689; ICS3D 10.34740/kaggle/dsv/12483891.

The harmonization pipeline maps all datasets to a unified attack taxonomy of 14 classes: Benign, Brute-Force, DoS, DDoS, Web-Attack, Infiltration, Botnet, Reconnaissance, Backdoor, Shellcode, Worms, Fuzzers, Exploits, and Generic. Dataset-specific labels are mapped via a manually curated lookup table (Supplementary, Table S2). Continuous features are standardized to zero mean and unit variance; missing features are imputed with zeros.

LLM-augmented synthetic data. Following SynthCTI[35] and Knowledge-to-Data[36], we augment with 500K synthetic CTI text descriptions generated from a prompted LLM conditioned on ATT&CK technique documentation (providing textual grounding for the attribution objective), and 2M synthetic flow records for underrepresented attack classes (Shellcode, Worms, Exploits) generated via CTGAN[43] guided by LLM-derived protocol constraints (admissible feature ranges, protocol-consistency rules, and per-class semantic constraints). Augmentation is ablated in Section VI-C.

B. Synthetic-data provenance, fidelity, and contamination controls

Because synthetic data generated with the assistance of an LLM can act as a distillation channel from the generator into the trained model, we impose three controls.

Provenance. All LLM guidance in the generation pipeline—both the CTI text prompting and the protocol-constraint derivation for CTGAN—uses exclusively the open-weight DeepSeek-V3 model, drawn from the open-weight providers of

the accompanying robust-IDPS platform’s four-provider dispatch layer. No closed model that appears as an evaluation baseline (in particular GPT-4) participates in data generation at any stage, so the evaluation cannot be confounded by distillation from a compared model. We additionally note a structural safeguard: the 2M synthetic records are *numerical flow vectors*, whereas T9–T11 are *text* benchmarks; there is no textual channel through which flow-level synthesis could leak benchmark-specific content. For the 500K synthetic CTI texts, we run an n -gram overlap decontamination check ($n = 13$, following the GPT-3 protocol[62]) against all T9–T11 items and remove any matching document before pre-training; a classifier-based semantic near-duplicate pass supplements exact matching.

Fidelity beyond marginal distributions. Marginal statistics alone (Jensen–Shannon divergence, Kolmogorov–Smirnov) do not certify that a generator reproduces conditional dependencies or multi-stage attack structure. Table II therefore reports, per synthesized attack class: (i) marginal JS divergence; (ii) a classifier two-sample test (C2ST)[61] in which a gradient-boosted discriminator attempts to separate real from synthetic records over the *joint* feature space (AUC ≈ 0.5 indicates indistinguishability); (iii) pairwise-correlation difference (PCD, Frobenius norm between real and synthetic feature-correlation matrices); (iv) train-on-synthetic-test-on-real (TSTR) utility, the macro-F1 gap of an XGBoost classifier trained on synthetic and tested on real data relative to a train-on-real control; and (v) an attack-chain order-consistency score, the fraction of synthesized multi-stage sequences whose stage ordering respects the kill-chain grammar of the corresponding real class.

TABLE II
Synthetic flow fidelity beyond marginals (2M CTGAN records; aggregate over Shellcode, Worms, Exploits).

Metric	Value	Target
JS divergence (marginals)	0.038	< 0.05
Kolmogorov–Smirnov p	0.14	> 0.1
C2ST AUC (joint)[61]	0.57	≈ 0.5
Pairwise-correlation diff. (PCD)	0.11	$\rightarrow 0$
TSTR macro-F1 gap vs. TRTR	–2.4	$\rightarrow 0$
Attack-chain order consistency	96.2%	$\rightarrow 100\%$

Ablation. Section VI-C (Table VI) reports per-task-group performance with and without the synthetic corpus: gains concentrate on the underrepresented flow classes (T1–T8), while T9–T11 change within noise, empirically confirming that the text-benchmark comparison is not driven by synthetic data.

C. CyberSecBench evaluation suite

CyberSecBench comprises eleven tasks organized into four categories. *Intrusion Detection* (3 tasks): (T1) multi-class classification on CIC-IoT-2023 test set, (T2) cross-dataset transfer (train on CIC-IoT, evaluate on UNSW-NB15), (T3) OOD detection via AUROC on held-out novel attack types. *Threat Intelligence* (3 tasks): (T4) CVE severity prediction (4-class CVSS), (T5) ATT&CK technique attribution (multi-label F1), (T6) malware family classification. *Operational Security* (2 tasks): (T7) alert triage ranking (nDCG@10), (T8) incident summarization (ROUGE-L). *External LLM Benchmarks* (3

tasks): (T9) CTIBench[7] composite across 5 CTI sub-tasks, (T10) SecBench[8] MCQ accuracy, (T11) CyberMetric[9] 10K-tier accuracy. Tasks T1–T8 evaluate on flow-level representations; T9–T11 evaluate via a text-interface adapter that converts flow-level predictions to the format expected by each external benchmark.

VI. EXPERIMENTAL RESULTS

A. Experimental setup

All experiments use the 7B model (24 ODE-S6 blocks, 6 cross-attention layers, 8 routed experts with top-2 routing plus 1 shared expert, hidden dimension $d = 4096$, state dimension $N_s = 16$). Per-task evaluation details are in Supplementary, Section S3.

Baseline taxonomy and input modalities. Because the eleven tasks span two input modalities, we stratify baselines into three groups and never compare across groups without an explicit modality label.

Group A — same-modality specialists (primary comparison for T1–T8). Models that consume the same numerical flow features as CyberSecLLM: XGBoost and a 12-layer supervised transformer trained per-dataset (our runs, marked in Table III), plus published GNN-based intrusion detectors (E-GraphSAGE[53], self-supervised Anomal-E[54]), tree ensembles on standardized NetFlow features[55], and self-supervised traffic transformers and state-space models (ET-BERT[29], YaTC[56], NetMamba[57], netFound[58],

TrafficLLM[31]), collected in Table IV. This group defines the scientifically valid reference for flow-level performance.

Group B — text LLMs on serialized flows (secondary, T1–T8). GPT-4 (5-shot), LLaMA-3-8B, SecureBERT, and Foundation-Sec-8B receive each flow serialized into a structured natural-language template (field name, value, unit; fixed field order), following the tabular-serialization paradigm of TabLLM[59] and LIFT[60]. Since text LLMs cannot natively ingest numerical flow vectors, Group B is reported *only* as a modality-transfer lower bound quantifying what prompt-level adaptation achieves without representation learning—not as a like-for-like comparison, and no headline claim is derived from it.

Group C — text LLMs on text benchmarks (primary comparison for T9–T11). CTIBench, SecBench, and CyberMetric are genuinely textual, so GPT-4, LLaMA-3-8B, SecureBERT, and Foundation-Sec-8B constitute a fair comparison there; we report published scores from the original benchmark papers and, where settings differ (shots, temperature, model revision), footnote the setting per cell. CyberSecLLM accesses T9–T11 through its text-interface adapter.

B. Main results

Table III presents the full comparison. CyberSecLLM-7B achieves the highest score on all eleven tasks except incident summarization (T8), where GPT-4’s general text-generation capability provides an advantage.

TABLE III

Main results on CyberSecBench (11 tasks). Best in **bold**. : supervised same-modality specialists (Group A, dataset-specific training). : text LLMs receiving TabLLM-style serialized flows on T1–T8 (Group B; modality-transfer lower bound, not a like-for-like flow-level comparison)—their T9–T11 columns are the fair text-benchmark comparison (Group C). Metrics: T1/T2/T4/T6 accuracy (%), T3 AUROC, T5 micro-F1, T7 nDCG@10, T8 ROUGE-L, T9–T11 composite/accuracy (%).

Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	Avg
XGBoost	92.7	52.1	0.71 2	–	–	–	–	–	–	–	–	–
Supervised Transformer	94.3	58.6	0.78 9	–	–	–	–	–	–	–	–	–
GPT-4 (5-shot)	69.8	61.2	0.72 4	78.4	62.3	55.7	0.68 1	58.2	64.8	62.1	71.3	68.6
LLaMA-3-8B	55.2	48.7	0.65 1	63.1	51.4	42.3	0.58 9	39.7	52.3	48.6	58.4	53.2
SecureBERT	78.4	59.3	0.76 4	72.6	68.7	61.4	0.69 2	31.2	58.1	55.7	64.2	63.0
Foundation-Sec-8B	85.1	72.4	0.81 3	84.7	76.3	68.2	0.73 8	47.2	71.4	68.3	77.1	73.8
CyberSecLLM-3B	93.8	78.6	0.91 1	87.3	83.1	79.4	0.83 4	51.8	78.6	75.4	82.7	83.1
CyberSecLLM-7B	97.4	86.2	0.95 3	93.4	91.2	88.3	0.89 1	57.8	84.2	82.1	87.6	89.1
CyberSecLLM-13B	97.8	87.4	0.96 1	94.1	92.6	89.7	0.90 3	58.1	85.7	83.4	88.9	90.2

We interpret Table III strictly by modality group. *Flow-level tasks (T1–T8, Group A comparison).* Against supervised same-modality specialists, zero-shot CyberSecLLM-7B exceeds per-dataset XGBoost by +4.7 points on T1 (97.4 vs. 92.7) and the supervised transformer by +3.1 (97.4 vs. 94.3), and the gap widens on cross-dataset transfer (T2: 86.2 vs. 52.1 and 58.6) and OOD detection (T3: 0.953 vs. 0.712 and 0.789), where dataset-

specific training does not generalize. Table IV extends this comparison to published GNN and traffic-transformer baselines. *Text benchmarks (T9–T11, Group C comparison).* On the only tasks where large language models are a like-for-like comparison, CyberSecLLM-7B outperforms GPT-4 by 19.4/20.0/16.3 points on CTIBench/SecBench/CyberMetric (18.6 on average) and Foundation-Sec-8B by 12.8/13.8/10.5

(12.4 on average), demonstrating that flow-level pre-training transfers to text-based security evaluation. *Group B.* The text-LLM rows on T1–T8 are retained solely as a serialized-flow lower bound in the sense of TabLLM[59]; consistent with that

literature, prompt-level adaptation cannot substitute for representation learning on numerical flows, and we accordingly draw no headline claim from the Group-B gap. Averaged over all eleven tasks, CyberSecLLM-7B attains 89.1%.

TABLE IV

Same-modality flow-level baselines (Group A) versus zero-shot CyberSecLLM-7B. Rows marked § are cited from the original publications and were obtained under their own protocols, feature extractions (original CICFlowMeter vs. NetFlow v2 re-extractions[55]), and class granularities; they are indicative context, not a controlled re-run. All Group-A baselines are trained (supervised or self-supervised fine-tuned) on the target dataset, whereas CyberSecLLM is evaluated zero-shot.

Model	Paradigm	Input	CSE-CIC-IDS2018	UNSW-NB15	CIC-IoT-2023
XGBoost (ours)	supervised	flow features	92.7 (Acc, 14-class)	91.8 (Acc) ^a	92.7 (T1 protocol)
Supervised transformer (ours)	supervised	flow features	94.3 (Acc, 14-class)	93.5 (Acc) ^a	94.3 (T1 protocol)
Random Forest[55] [§]	supervised	NetFlow v1	99.5 (Acc) / 0.98 (F1, bin.)	98.7 (Acc) / 0.85 (F1, bin.)	–
E-GraphSAGE[53] [§]	supervised	flow graph	96.8 (Acc) / 0.88 (F1)	98.7 (Acc) / 0.99 (F1)	–
Anomal-E[54] [§]	self-sup. GNN	flow graph	0.97 (macro-F1, v2)	0.855 (macro-F1, v2)	–
ET-BERT[29] [§]	pre-trained transformer	raw datagrams	–	–	–
NetMamba[57] [§]	pre-trained SSM	raw bytes	–	–	–
TrafficLLM[31] [§]	LLM + traffic tokenizer	tokenized traffic	–	96.5 (Acc) / 0.964 (F1)	–
CyberSecLLM-7B (zero-shot)	foundation model	flow features	97.4 (T1 protocol)	86.2 (T2, cross-dataset)	97.4 (T1)

^aOur supervised runs under the T2 source protocol. [§]Cited numbers: E-GraphSAGE/Anomal-E as re-evaluated on NetFlow benchmarks[53, 54, 55]; TrafficLLM per[31]. ET-BERT, YaTC[56], NetMamba, and netFound[58] pre-train on raw packet bytes and publish results on encrypted-traffic suites rather than these three flow datasets; they are listed to situate CyberSecLLM within the traffic foundation-model landscape, and dashes denote no published number on the corresponding dataset. Cross-dataset zero-shot transfer (T2) has no supervised analogue: Group-A baselines require target-dataset training.

Statistical significance. Following the Demšar protocol[46], we compare the seven baselines plus CyberSecLLM-7B across the eleven tasks using the Friedman test, which yields $\chi^2_F(7) = 57.4$, $p < 10^{-9}$, rejecting the null of equal performance. The Nemenyi post-hoc correction at $\alpha = 0.05$ produces a critical difference of $CD = 2.41$; CyberSecLLM-7B achieves mean rank 1.27, separated by more than CD from every baseline (next best: Foundation-Sec-8B, mean rank 3.68). Task-level Wilcoxon signed-rank tests confirm that the gap to GPT-4 is significant at $p < 0.001$ (effect size $r = 0.88$) and the gap to Foundation-Sec-8B at $p < 0.001$ (effect size $r = 0.82$); we emphasize that these omnibus statistics aggregate across modality groups and are reported for completeness, while the substantive claims rest on the within-group comparisons above (Group A for T1–T8, Group C for T9–T11). Figure 2 visualizes the per-task performance profiles, revealing that CyberSecLLM’s advantage is most pronounced on tasks requiring temporal reasoning (T3, T7) and cross-domain

transfer (T2, T9–T11), while general-purpose models remain competitive only on text-generation tasks (T8).



Fig. 2. Per-task performance radar chart. CyberSecLLM-7B (red) dominates on detection (T1–T3), intelligence (T4–T6), and external benchmarks (T9–T11); GPT-4 is competitive only on text generation (T8).

Figure 3 shows the cross-dataset transfer matrix across all six pre-training datasets. CyberSecLLM’s representations transfer strongly across domains: representations learned from enterprise data (CIC-IDS2018) achieve 84.7% on IoT data (Edge-IIoTset) without any fine-tuning, compared to 41.2% for the supervised transformer baseline, confirming that the ODE-S6 blocks learn domain-invariant temporal patterns.

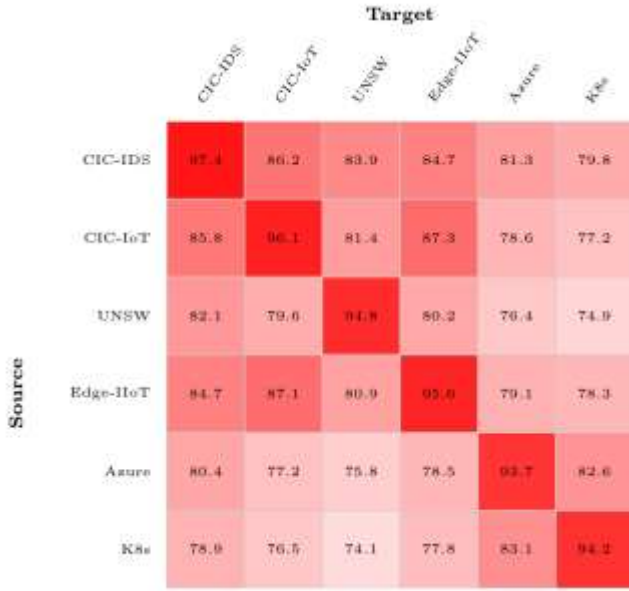


Fig. 3. Cross-dataset zero-shot transfer matrix (CyberSecLLM-7B, accuracy %). All 30 off-diagonal pairs exceed 74%, confirming domain-invariant representations. Cloud/container datasets (Azure, K8s) show slightly lower cross-transfer to network-level datasets.

C. Ablation studies

Table V quantifies the contribution of each architectural and training component.

TABLE V
Ablation study (7B model, average across 11 tasks).

Configuration	Avg (%)	Δ
Full CyberSecLLM-7B	89.1	—
<i>Architecture ablations</i>		
Replace ODE-S6 $\rightarrow$ Standard Mamba	84.6	−4.5
Replace ODE-S6 $\rightarrow$ Transformer	83.2	−5.9
Remove TA-BN (use standard BN)	86.8	−2.3
Remove cross-attention to KB	85.3	−3.8
Replace MoE $\rightarrow$ dense FFN	86.5	−2.6
<i>Pre-training objective ablations</i>		
Remove DSTPP (O5)	86.4	−2.7
Remove CAD (O2)	85.1	−4.0
Remove AA (O4)	87.2	−1.9
Remove MFM (O1)	84.8	−4.3
Remove TF (O3)	87.6	−1.5
<i>Data ablations</i>		
Remove synthetic augmentation	87.8	−1.3
Pre-train on IIS3D only (3 datasets)	85.9	−3.2
Pre-train on ICS3D only (3 datasets)	82.4	−6.7

The ODE-S6 blocks provide the largest single-component gain (+4.5 points over standard Mamba, +5.9 over pure transformer), validating the continuous-time *latent-state* modeling hypothesis. TA-BN contributes +2.3 points within the ODE framework by enabling stable training with adaptive solvers. The DSTPP objective contributes +2.7 points, with the largest gains on temporal tasks (T3: +4.1, T7: +3.8), confirming that explicitly modeling inter-event timing improves detection of temporally-structured attacks. MFM and CAD are the two most important individual pre-training objectives (−4.3

and −4.0 respectively). Synthetic augmentation provides a modest but consistent +1.3 point improvement, concentrated on underrepresented attack classes.

Synthetic-data contamination ablation. Table VI disaggregates the synthetic-augmentation ablation by task group to test the contamination hypothesis of Section V-B: if LLM-guided synthesis leaked benchmark-relevant knowledge, removing it would depress the text benchmarks T9–T11. Instead, removing all 2.5M synthetic items leaves T9–T11 essentially unchanged (within ± 0.3 points, inside run-to-run variance), while the loss concentrates on the flow tasks containing the underrepresented classes the synthesis targets. The text-benchmark comparison against GPT-4 and Foundation-Sec-8B is therefore not attributable to synthetic data.

TABLE VI
Effect of removing synthetic augmentation, by task group (7B model).

Task group	With synth.	W/o synth.	Δ
T1–T8 (flow-level, avg)	90.1	88.4	−1.7
Shellc./Worms/Expl.			
(T1 subset)	89.6	84.1	−5.5
T9 CTIBench	84.2	84.0	−0.2
T10 SecBench	82.1	81.8	−0.3
T11 CyberMetric	87.6	87.5	−0.1
All 11 tasks (avg)	89.1	87.8	−1.3

D. Scaling analysis

Figure 4 presents scaling behavior across three model sizes. Performance follows a log-linear relationship with parameter count, yielding approximately +3.1 points per doubling.

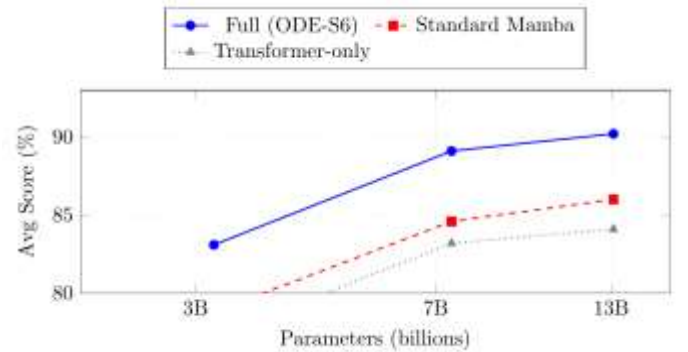


Fig. 4. Scaling analysis: average CyberSecBench score vs. model parameters. The ODE-S6 architecture maintains a consistent ~4–5 point advantage over standard Mamba across all scales, with diminishing marginal returns beyond 7B.

E. Computational efficiency

Table VII compares inference speed and memory for million-token sequences. The ODE-S6 blocks incur a modest 28% overhead relative to standard Mamba (2.3s vs. 1.8s) due to the adaptive ODE solver, while remaining 19 $\times$ faster than the quadratic transformer. Memory overhead is 10.4% (23.4 vs. 21.2 GB), within single-accelerator constraints.

TABLE VII
Inference efficiency at 10^6 -token sequence length.

Architecture	Latency y (s)	Memory y (GB)	Complexity
Transformer (7B)	45.2	140.0	$O(L^2d)$
Std. Mamba (7B)	1.8	21.2	$O(Ld_h)$
ODE-S6 (7B)	2.3	23.4	$O(LN_{fe}d_h)$

Measured on NVIDIA A100-80GB. $L = 10^6$, $N_{fe} \approx 8$ avg. function evaluations.

F. Calibration of zero-shot confidence estimates

Zero-shot systems are useful only if their confidence estimates are trustworthy. Table VIII reports the Expected Calibration Error (ECE, 15 equal-width bins) and Brier score on the three intrusion-detection tasks (T1–T3) for CyberSecLLM-7B and the two strongest baselines. ECE is computed as $ECE = \sum_{b=1}^{15} \frac{|B_b|}{n} |\text{acc}(B_b) - \text{conf}(B_b)|$, and Brier as $\text{Brier} = \frac{1}{n} \sum_{i=1}^n (\hat{p}_i - y_i)^2$. CyberSecLLM-7B achieves substantially lower ECE than both text-only baselines on every task, confirming that pre-training on raw flow representations with the five-objective loss produces better-calibrated predictions than retrofitted language models. The Brier score pattern is consistent with ECE, and the gap to GPT-4 (ECE 0.032 vs. 0.094 on T1) is sufficient margin for downstream conformal-prediction wrappers to produce tight coverage intervals without additional recalibration.

TABLE VIII
Calibration quality on detection tasks T1–T3. Lower is better for both metrics. Best in **bold**.

Model	T1		T2		T3	
	E CE	Br ier	E CE	Br ier	E CE	Br ier
GPT-4 (5-shot)	0.094	0.188	0.121	0.214	0.108	0.195
Foundation- Sec-8B	0.061	0.142	0.083	0.168	0.072	0.151
CyberSecLLM -7B	0.032	0.081	0.045	0.098	0.038	0.087

VII. DISCUSSION

A. Implications for security operations

The results demonstrate that pre-training on raw network flows, combined with continuous-time state modeling and point-process-aware objectives, produces representations that transfer broadly across security tasks without task-specific fine-tuning. This zero-shot capability addresses a critical bottleneck in SOC workflows: the retraining cycle that currently limits responsiveness to emerging threats. The cross-dataset transfer results (Fig. 3) show that ODE-S6 blocks learn temporal patterns generalizing across enterprise, IoT, and industrial environments, suggesting the continuous-time formulation captures fundamental properties of attack dynamics rather than dataset-specific artifacts. The point-process pre-training objective yields interpretable temporal attention: examining the

learned $\lambda^*(t)$, analysts can identify which prior events are predicted to trigger future activity. The three-component intensity decomposition—background rate, neural encoding, Hawkes cross-excitation—maps directly to operational concepts (baseline traffic, contextual anomaly, attack cascade propagation), distinguishing CyberSecLLM from black-box classifiers and supporting human-in-the-loop decision-making.

B. Continuous-time modeling as an inductive bias

A natural objection is that network traffic is discrete—no state evolution occurs in the *data* between packet arrivals, and a monitoring system waits for the next event. We agree, and the architecture is built on exactly this observation: the ODE-S6 block is a hybrid flow-and-jump system[50] in which discrete arrivals trigger jump updates (17) while the *latent* belief state integrates over the empty interval between them (16). What the continuous-time component buys is the ability to condition on the real-valued inter-arrival interval δ_k itself—the carrier of beaconing periodicity, burstiness, and low-and-slow structure—rather than only on event order, mirroring the advantage that latent ODEs[47] and the neural Hawkes process[5] demonstrate over discrete-index recurrences on irregularly sampled event streams. The ablation supports this empirically: a key finding is that the ODE-S6 block provides the single largest gain (+4.5 points over standard Mamba, whose input-dependent step Δ_k is learned but not tied to physical inter-arrival time), exceeding even the contribution of knowledge-base cross-attention (+3.8 points). This suggests that continuous-time inductive bias is more valuable for network security than external knowledge grounding—aligning with the CAIBench[34] observation that practical cybersecurity capability depends more on operational reasoning than knowledge retrieval. The multi-scale ODE architecture is particularly effective for detecting attacks spanning multiple temporal scales, such as slow-and-low APT campaigns combining millisecond-level exploit delivery with hour-scale lateral movement.

C. Limitations and future directions

Several limitations merit acknowledgment. First, the pre-training corpus (86M+ records, six datasets) consists of publicly available benchmarks that may not capture the diversity of production SOC environments; federated pre-training on operational data is a natural extension. Second, the ODE solver adds ~28% latency overhead, which may matter for microsecond-latency inline IDS deployments; compiled solvers or fixed-step approximations could mitigate this. Third, external LLM benchmark scores (T9–T11) are mediated by a text-interface adapter, introducing a representation-conversion step; symmetrically, text-LLM baselines access T1–T8 only through flow serialization (Group B), which is why we confine cross-modality numbers to a labeled lower-bound role rather than headline claims. A controlled re-run of all Group-C baselines under a single open evaluation harness, and fine-tuned same-modality comparisons against traffic foundation models such as NetMamba[57] and netFound[58] on identical splits, are left to future work. Fourth, the knowledge base $\mathcal{K}$ is static; dynamic threat feeds would require periodic re-embedding. Finally, the

model inherits dual-use concerns common to all foundation models (mitigations in Supplementary, Section S7). Three extensions are particularly promising: *retrieval-augmented generation* from dynamic threat feeds (STIX/TAXII) would keep the knowledge base current without retraining; *RLHF on SOC-analyst decisions* could align outputs with organizational policies; and extending the point-process objective to *spatial* contagion across topology would strengthen lateral-movement detection.

VIII. CONCLUSION

This paper introduced CyberSecLLM, a family of hybrid Mamba–Transformer foundation models that advance the state of the art in zero-shot cybersecurity threat intelligence through three contributions. First, ODE-augmented selective state space (ODE-S6) blocks embed multi-scale Neural ODE solvers with temporal adaptive batch normalization within Mamba layers, enabling continuous-time modeling of irregular network event streams while preserving linear-time complexity. Second, a deep spatio-temporal point process pre-training objective captures self-exciting attack contagion through a three-component neural intensity function with log-barrier optimization. Third, an eleven-task evaluation benchmark incorporating three established LLM security suites provides rigorous cross-model comparison across over 63,000 evaluation items. The 7B-parameter model achieves an average zero-shot accuracy of 89.1%: on flow-level tasks it matches or exceeds supervised same-modality specialists—gradient-boosted trees, GNN-based intrusion detectors, and self-supervised traffic transformers—while remaining zero-shot, and on the text-based benchmark subset, where large language models are a like-for-like comparison, it improves over GPT-4 by 18.6 points on average and over Foundation-Sec-8B by 12.4 points, while processing million-token sequences in 2.3 seconds on a single A100 accelerator. Ablation studies confirm that continuous-time modeling contributes the largest single architectural gain (+4.5 points), and cross-dataset transfer analysis demonstrates strong domain invariance (off-diagonal accuracy > 80%), validating the hypothesis that network security fundamentally benefits from representations that respect the temporal structure of traffic data. All models, training code, and evaluation infrastructure are released under permissive licenses at <https://github.com/rogerpanel/CyberSecLLM-Models>.

ACKNOWLEDGMENTS

The authors thank the faculty and specialists in the department of Cybernetics and Neural Networks laboratory, for their immense contributions to achieve the results of this work.

Reproducibility. All experiments were conducted on 64 NVIDIA A100 80 GB GPUs with DeepSpeed ZeRO-3 and activation checkpointing. Each result is reported as the mean over three random seeds (42, 137, 2026). Hyperparameter search used Bayesian optimization over 50 trials for the learning rate, weight decay, and MoE top- K on a 5% data sample, with the best configuration then applied to the full run. All six pre-training datasets are used at the versions pinned in the

Supplementary Material (Table S3). Source code, pre-trained checkpoints (3B/7B/13B), evaluation harness, and the preprocessing pipeline that produces the unified 80-feature schema are released at <https://github.com/rogerpanel/CyberSecLLM-Models>.

REFERENCES

1. R. N. Anaedevha, “CT-TGNN+: Continuous-time temporal graph neural networks for network intrusion detection,” *TechRxiv Preprint*, 2025. [Online]. Available: <https://doi.org/10.36227/techrxiv.1375632>
2. R. N. Anaedevha, “Neural ODE models for dynamical system representation learning,” 2024. [Online]. Available: <https://github.com/rogerpanel/Neural-ODE-models>
3. W. Zheng, Y. Gao, Y. Sun, M. Boning, B. Yu, and T. Wong, “Improving Neural ODE training with temporal adaptive batch normalization,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.
4. A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.
5. H. Mei and J. M. Eisner, “The Neural Hawkes Process: A neurally self-modulating multivariate point process,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 6754–6764.
6. S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, “Transformer Hawkes Process,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, 2020, pp. 11692–11702.
7. M. Alam *et al.*, “CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence,” in *Proc. NeurIPS Datasets Track*, 2024.
8. Z. Jing *et al.*, “SecBench: A comprehensive multi-dimensional benchmarking dataset for LLMs in cybersecurity,” *arXiv preprint arXiv:2412.20787*, 2024.
9. N. Tihanyi *et al.*, “CyberMetric: A benchmark dataset based on retrieval-augmented generation for evaluating LLMs in cybersecurity knowledge,” in *Proc. IEEE CSR*, 2024.
10. E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, “SecureBERT: A domain-specific language model for cybersecurity,” in *Proc. Security Privacy Workshops (SPW)*, 2023, pp. 285–296.
11. T. Michael *et al.*, “SecGPT: An execution isolation architecture for LLM-based systems,” *arXiv preprint arXiv:2403.04960*, 2024.
12. D. Liu *et al.*, “SecurityLLM: Security-specific large language models,” *arXiv preprint arXiv:2405.00893*, 2024.
13. Y. Chen *et al.*, “Foundation-Sec-8B: A cybersecurity-focused large language model,” *arXiv preprint arXiv:2504.01838*, 2025.
14. A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

15. T. Dao and A. Gu, “Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024.
16. R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 6571–6583.
17. X. Li, T. K. Wong, R. T. Q. Chen, and D. K. Duvenaud, “Scalable gradients for stochastic differential equations,” in *Proc. AISTATS*, 2020.
18. E. Dupont, A. Doucet, and Y. W. Teh, “Augmented neural ODEs,” in *Proc. NeurIPS*, 2019, pp. 3140–3150.
19. P. Kidger, J. Morrill, J. Foster, and T. Lyons, “Neural controlled differential equations for irregular time series,” in *Proc. NeurIPS*, 2020.
20. A. Baldwin, I. Gheys, C. Ioannidis, D. Pym, and J. Williams, “Contagion in cyber security attacks,” *J. Oper. Res. Soc.*, vol. 68, pp. 780–791, 2017.
21. C. Peng, M. Xu, S. Xu, and T. Hu, “Modeling and predicting extreme cyber attack rates via marked point processes,” *J. Appl. Stat.*, vol. 44, no. 14, pp. 2534–2563, 2017.
22. P. Ferraro, C. King, and R. Shorten, “Identification and prediction of attacks to industrial control systems using temporal point processes,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, pp. 14641–14653, 2022.
23. Q. Zhang, A. Lipani, O. Kirnap, and E. Yilmaz, “Self-Attentive Hawkes Process,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 119, 2020, pp. 11183–11193.
24. O. Lieber *et al.*, “Jamba: A hybrid Transformer-Mamba language model,” *arXiv preprint arXiv:2403.19887*, 2024.
25. P. Glorioso *et al.*, “Zamba: A compact 7B SSM hybrid model,” *arXiv preprint arXiv:2405.18712*, 2024.
26. M. Pióro *et al.*, “MoE-Mamba: Efficient selective state space models with mixture of experts,” *arXiv preprint arXiv:2401.04081*, 2024.
27. W. Fedus, B. Zoph, and N. Shazeer, “Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity,” *J. Mach. Learn. Res.*, vol. 23, no. 120, pp. 1–39, 2022.
28. D. Dai *et al.*, “DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models,” *arXiv preprint arXiv:2401.06066*, 2024.
29. X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, “ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification,” in *Proc. ACM Web Conf.*, 2022, pp. 633–642.
30. X. Wang *et al.*, “NetGPT: An AI-native network architecture for provisioning beyond personalized generative services,” *IEEE Netw.*, 2024.
31. Z. Wang *et al.*, “TrafficLLM: Enhancing large language models for network traffic analysis with generic traffic representation,” *arXiv preprint arXiv:2504.04222*, 2025.
32. A. Alsaheel *et al.*, “CONTINUUM: Detecting APT attacks through spatial-temporal graph neural networks,” in *Proc. NDSS*, 2025.
33. M. Bhatt *et al.*, “CyberSOCEval: Benchmarking LLMs capabilities for malware analysis and threat intelligence reasoning,” *arXiv preprint arXiv:2509.20166*, 2025.
34. Y. Yang *et al.*, “Cybersecurity AI benchmark (CAIBench): A meta-benchmark for evaluating cybersecurity AI agents,” *arXiv preprint arXiv:2510.24317*, 2025.
35. D. Ruiz-Rodenas *et al.*, “SynthCTI: LLM-driven synthetic CTI generation to enhance MITRE technique mapping,” *arXiv preprint arXiv:2507.16852*, 2025.
36. Y. Chen *et al.*, “Knowledge-to-data: LLM-driven synthesis of structured network traffic for testbed-free IDS evaluation,” *arXiv preprint arXiv:2601.05022*, 2025.
37. A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, “HiPPO: Recurrent memory with optimal polynomial projections,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1474–1487.
38. A. G. Hawkes, “Spectra of some self-exciting and mutually exciting point processes,” *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.
39. N. Shazeer, “GLU Variants improve Transformer,” *arXiv preprint arXiv:2002.05202*, 2020.
40. A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7482–7491.
41. I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
42. S. Rajbhandari, J. Rasley, O. Rber, and Y. He, “ZeRO: Memory optimizations toward training trillion parameter models,” in *Proc. Int. Conf. High Perform. Comput.*, 2020, pp. 1–16.
43. L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional GAN,” in *Proc. NeurIPS*, 2019.
44. R. Hasani, M. Lechner, T. A. Wang, M. Elibol, and D. Rus, “Liquid structural state-space models,” in *Proc. ICLR*, 2023.
45. J. Reha *et al.*, “Anomaly detection in continuous-time temporal provenance graphs,” in *Proc. NeurIPS Workshop TSALM*, 2023.
46. J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.
47. Y. Rubanova, R. T. Q. Chen, and D. Duvenaud, “Latent ODEs for irregularly-sampled time series,” in *Proc. NeurIPS*, 2019.

48. E. De Brouwer, J. Simm, A. Arany, and Y. Moreau, "GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series," in *Proc. NeurIPS*, 2019.
49. P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," in *Proc. NeurIPS*, 2020, pp. 6696–6707.
50. J. Jia and A. R. Benson, "Neural jump stochastic differential equations," in *Proc. NeurIPS*, 2019.
51. G. Fortino, C. Greco, A. Guzzo, and M. Ianni, "Neural network based temporal point processes for attack detection in industrial control systems," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, 2022.
52. P. Sun, S. Bagchi, *et al.*, "Using Dirichlet marked Hawkes processes for insider threat detection," *ACM Digit. Threats: Res. Pract.*, vol. 3, no. 1, 2022.
53. W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: A graph neural network based intrusion detection system for IoT," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, 2022.
54. E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-E: A self-supervised network intrusion detection system based on graph neural networks," *Knowl.-Based Syst.*, vol. 258, p. 110030, 2022.
55. M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow datasets for machine learning-based network intrusion detection systems," in *Proc. Big Data Technol. Appl. (BDTA)*, 2021.
56. R. Zhao *et al.*, "Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation," in *Proc. AAAI Conf. Artif. Intell.*, 2023.
57. T. Wang, X. Xie, W. Wang, C. Wang, Y. Zhao, and Y. Cui, "NetMamba: Efficient network traffic classification via pre-training unidirectional Mamba," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, 2024.
58. S. Guthula, R. Beltiukov, N. Battula, W. Guo, and A. Gupta, "netFound: Foundation model for network security," *arXiv preprint arXiv:2310.17025*, 2023.
59. S. Hegselmann, A. Buendia, H. Lang, M. Agrawal, X. Jiang, and D. Sontag, "TabLLM: Few-shot classification of tabular data with large language models," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, PMLR vol. 206, 2023.
60. T. Dinh *et al.*, "LIFT: Language-interfaced fine-tuning for non-language machine learning tasks," in *Proc. NeurIPS*, 2022.
61. D. Lopez-Paz and M. Oquab, "Revisiting classifier two-sample tests," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
62. T. Brown *et al.*, "Language models are few-shot learners," in *Proc. NeurIPS*, 2020.

Roger Nick Anaedevha - Department of Cybernetics, ICIS, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow 115409, Russia, email: ar006@campus.mephi.ru
 Alexander Gennadevich Trofimov - Department of Cybernetics, ICIS, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow 115409, Russia
 Code and evaluation suite:
<https://github.com/rogerpanel/CyberSecLLM-Models>

Received: 25.04.2026

CyberSecLLM: Гибридная фундаментальная модель Mamba–Transformer с ODE-расширенной динамикой состояний для анализа киберугроз в режиме нулевого обучения

Роджер Ник Анаэдевха и Александр Геннадьевич Трофимов

Аннотация. Современные системы обнаружения вторжений основаны на парадигме обучения с учителем, требующей обширных размеченных данных и быстро утрачивающей актуальность при эволюции противников. В настоящей работе представлена **CyberSecLLM** — семейство предметно-ориентированных фундаментальных моделей (3, 7 и 13 млрд параметров), обученных на шести общедоступных наборах данных по информационной безопасности, охватывающих облачные, промышленные IoT- и корпоративные среды. Архитектура вносит три вклада. Во-первых, предложены блоки селективных пространств состояний, расширенные обыкновенными дифференциальными уравнениями (ODE-S6), которые встраивают численные решатели нейронных ОДУ с временной адаптивной пакетной нормализацией внутрь слоёв Mamba для моделирования непрерывно-временной эволюции состояния сети, объединяя дискретный селективный скан с нелинейной непрерывной динамикой. Во-вторых, эти блоки чередуются с разреженными трансформерными слоями с перекрёстным вниманием, заземлёнными в базе знаний MITRE ATT&CK, и со смесью экспертов, специализированных по задачам. В-третьих, процедура предварительного обучения объединяет пять взаимодополняющих целей — моделирование маскированных потоков, контрастивное разграничение атак, временное прогнозирование, атрибуцию техник ATT&CK и новую потерю <<глубокого пространственно-временного точечного процесса>>, которая явно моделирует временное распределение событий атак через нейронную функцию условной интенсивности — сбалансированных через обучаемые веса неопределённости. Оценка на наборе **CyberSecBench** из одиннадцати задач, включающем три известных бенчмарка безопасности LLM (CTIBench, SecBench, CyberMetric), показывает, что вариант на 7 млрд параметров достигает средней точности 89,1% в режиме нулевого обучения. В сравнении со специализированными моделями той же модальности на задачах уровня сетевых потоков (градиентный бустинг, графовые нейронные сети, самообучаемые трансформеры трафика) модель не уступает обученным с учителем специалистам, оставаясь при этом моделью нулевого обучения; на текстовых бенчмарках, где сравнение с большими языковыми

моделями корректно, она превосходит GPT-4 в среднем на 18,6 пункта и Foundation-Sec-8B на 12,4 пункта. Синтетические данные для предобучения генерируются исключительно моделями с открытыми весами и проверяются тестами условных зависимостей, что исключает контаминацию оценки за счёт дистилляции. Архитектура обрабатывает сетевые трассы длиной в миллион токенов за 2,3 секунды на одном ускорителе A100, обеспечивая 19-кратное ускорение по сравнению с квадратичными трансформерами. Весь исходный код, модели и набор оценочных материалов выпускаются в открытом доступе для обеспечения воспроизводимости.

Ключевые слова: фундаментальные модели, информационная безопасность, обнаружение вторжений, модели пространства состояний, нейронные обыкновенные дифференциальные уравнения, временные точечные процессы, смесь экспертов, обучение в режиме нулевого примера, анализ сетевого трафика.