Neurocomputing

Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale

--Manuscript Draft--

Manuscript Number:	NEUCOM-D-25-13536
Article Type:	Regular article
Section/Category:	Deep Learning
Keywords:	Adversarial Machine Learning; Gaussian Processes and Uncertainty Quantification; Multi-Scale Temporal Modeling and Adversarial Robustness, Network Intrusion Detection System; Heterogeneous Data and Cloud Security; Variational Inference and Domain Adaptation
Corresponding Author:	Roger Nick Anaedevha National Research Nuclear University MEPhI Moscow, RUSSIAN FEDERATION
First Author:	Roger Nick Anaedevha
Order of Authors:	Roger Nick Anaedevha
	Alexandre Gennadevich Trofimov
	Yuri V. Borodachev
Abstract:	Modern cloud environments face critical intrusion-detection challenges across heterogeneous domains (Edge-IIoT, containers, SOCs) with extreme class imbalance (up to 99:1). Deterministic deep models lack principled uncertainty, contributing to 10,000–50,000 daily alerts and double-digit false-positive rates. We present a hierarchical Gaussian Process (GP) framework that decomposes network behavior with specialized kernels for multi-scale temporal patterns spanning microseconds to weeks. Our variational sparse GP implementation scales per epoch as O(N_b M^2 + M^3) with mini-batch size N_b and provides O(M) per-point prediction. Evaluated on 21.48M ICS3D records, the method attains 96.5% accuracy and a 2.6% false-positive rate—a 42% relative reduction versus the best baseline ($p < 0.001$)—together with well-calibrated uncertainty (ECE < 0.038), enabling risk-aware alert prioritization, 68% analyst workload reduction, and improved robustness under adversarial perturbations.

October 12, 2025.

To: Editor-in-Chief, Neurocomputing

Subject: Submission to *Neurocomputing Journal*: "Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale Temporal Modeling"

Dear Editor,

We are pleased to submit our manuscript entitled "Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale Temporal Modeling" for consideration for publication in Neurocomputing.

Manuscript Overview

Modern cloud security systems face a critical challenge: handling 10,000-50,000 daily alerts while maintaining detection accuracy across heterogeneous environments. Current deep learning approaches achieve high accuracy but lack principled uncertainty quantification, resulting in alert fatigue that causes 35% of genuine incidents to be overlooked. Our work addresses this fundamental gap.

Key Contributions

This manuscript presents a novel hierarchical Gaussian Process framework with four major innovations:

1. *Hierarchical Architecture for Heterogeneity:* We introduce the first GP-based framework that unifies heterogeneous cloud security data (Edge-IIoT protocols, container microservices, SOC events) through decomposition into shared and domain-specific components with specialized kernels.
2. *Adversarially-Robust Sparse Approximation:* Our novel training procedure for inducing point selection maintains uncertainty calibration under adversarial perturbations, achieving 71.2% accuracy at $\varepsilon=0.1$ compared to 39.3% for Deep Ensembles while scaling as $O(N\_b\, M^2 + M^3)$ per epoch.
3. *Multi-Scale Temporal Modeling:* Domain-adaptive kernels capture attack patterns spanning seven orders of magnitude (microseconds to weeks), automatically discovering periodic components and change-points without manual feature engineering.
4. *Practical Impact:* Evaluated on 21.48M real-world records (ICS3D dataset), our method achieves 96.5% accuracy with 2.6% false positive rate—a 42% improvement over the best baseline (TranAD: 4.5% FPR)—while providing well-calibrated uncertainty (ECE < 0.038) that reduces analyst workload by 68%.

Significance and Novelty

This work makes three significant advances over existing literature:

Theoretical Rigor: Unlike recent uncertainty-aware deep learning approaches (Sensoy et al. 2018, Malinin & Gales 2018) that lack formal guarantees, our GP framework provides calibration proofs, PAC-Bayesian bounds, and convergence guarantees under stated assumptions.

Practical Scale: While previous GP applications to intrusion detection (Ramadas et al. 2003, Kim & Lee 2004) were limited to thousands of samples, our sparse variational approximation handles 21.48M heterogeneous records with real-time inference (7.8ms, 14K-28K events/s).

Operational Value: Beyond accuracy improvements, our uncertainty quantification enables intelligent alert prioritization: immediate response for high-confidence threats ($\sigma<1.0$), automated investigation for moderate uncertainty ($1.0<\sigma<2.0$), and batch review for low confidence ($\sigma>2.0$).

Fit with Neurocomputing

This manuscript aligns perfectly with Neurocomputing's scope and recent publications:

- *Machine Learning Theory:* Rigorous probabilistic framework with theoretical guarantees
- *Real-World Applications:* Addresses critical cybersecurity challenges with practical deployment
- *Computational Efficiency:* Novel sparse approximations enabling large-scale deployment
- *Related Publications:* Similar to recent Neurocomputing papers on uncertainty quantification (e.g., "Deep Gaussian Processes for..." by [Author, Year]) and heterogeneous learning

Reproducibility and Data Availability

Consistent with open science principles:

- The code Publicly available at: https://github.com/rogerpanel/CV/blob/54e68df602d75a2f4ce7143b24204c97b804ea66/hgp-model-v2.ipynb
- *Data:* ICS3D dataset available at Kaggle (DOI: 10.34740/kaggle/dsv/12483891) under CC BY-NC-SA 4.0 license
- *Supplementary Material:* Comprehensive 14-section appendix with all proofs, algorithms, and implementation details.

Manuscript Organization

The 29-page main manuscript follows a logical structure:

- Sections 1-2: Motivation and comprehensive related work (99 references)
- Section 3: Rigorous problem formulation with mathematical foundations
- Section 4: Hierarchical GP methodology with theoretical guarantees
- Section 5: Algorithms and implementation details
- Sections 6-7: Extensive experiments on 21.48M records, outperforming 15 baselines
- Sections 8-9: Discussion and conclusions with future directions

The supplementary appendix (14 sections) provides complete proofs, detailed kernel specifications, extended results, and three case studies demonstrating detection of zero-day MQTT attacks, container escapes, and APT campaigns.

Target Audience

This work will interest multiple Neurocomputing reader communities:

ML Researchers: Novel GP architecture and theoretical contributions
Security Practitioners: Practical framework for operational deployment
Applied Scientists: Case studies demonstrating real-world effectiveness
Method Developers: Transferable techniques for heterogeneous data

Declarations

- *Funding:* This work was supported by a grant from the Ministry of Economic Development of the Russian Federation (identifier 000000C313925P3Q0002).
- *Conflicts of Interest:* The authors declare no conflicts of interest.
- *Ethics Approval:* Not applicable (publicly available datasets, no human subjects).
- *Data Availability:* Fully available as detailed above.
- *Author Contributions:* R.N.A. designed the study, developed methods, conducted experiments, and wrote the manuscript. A.G.T. supervised the work and provided theoretical guidance. Y.V.B. contributed to experimental design and manuscript revision.
- *Previous Submission:* This manuscript has not been published elsewhere and is not under consideration at another journal.

We believe this manuscript makes significant theoretical and practical contributions to machine learning for cybersecurity, demonstrating how principled uncertainty quantification can address real operational challenges. The rigorous experimental validation on large-scale heterogeneous data, combined with theoretical guarantees and reproducible implementation, aligns well with Neurocomputing's standards for high-quality research.

Thank you for your consideration. We believe the manuscript will be of strong interest to Neurocomputing's readership, as we look forward to your editorial decision and reviewer feedback.

Sincerely,

RogerNickAnaedevha

Roger Nick Anaedevha (corresponding author)
National Research Nuclear University, MEPhI
Email: rogernickanaedevha@gmail.com | ar006.campus@mephi.ru

Alexander Gennadevich Trofimov
National Research Nuclear University MEPhI

Yuri Vladimirovich Borodachev
Artificial Intelligence Research Center, MEPhI

Abstract

Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale Temporal Modeling

Abstract

Modern cloud environments face critical intrusion-detection challenges across heterogeneous domains (Edge-IIoT, containers, SOCs) with extreme class imbalance (up to 99:1). Deterministic deep models lack principled uncertainty, contributing to 10,000–50,000 daily alerts and double-digit false-positive rates. We present a hierarchical Gaussian Process (GP) framework that decomposes network behavior with specialized kernels for multi-scale temporal patterns spanning microseconds to weeks. Our variational sparse GP implementation scales per epoch as $O(N_b M^2 + M^3)$ with mini-batch size N_b and provides $O(M)$ per-point prediction. Evaluated on 21.48M ICS3D records, the method attains 96.5% accuracy and a 2.6% false-positive rate—a 42% relative reduction versus the best baseline ($p < 0.001$)—together with well-calibrated uncertainty (ECE < 0.038), enabling risk-aware alert prioritization, 68% analyst workload reduction, and improved robustness under adversarial perturbations.

Keywords: Gaussian Processes, Uncertainty Quantification, Multi-Scale Temporal Modeling, Adversarial Robustness, Heterogeneous Data, Cloud Security, Variational Inference, Domain Adaptation

Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale Temporal Modeling

1. Introduction

Modern cloud infrastructure processes over 100 trillion daily requests across heterogeneous environments spanning Edge computing, IoT, containerized microservices, and Security Operations Centers (SOCs) [1]. The Integrated Cloud Security 3Datasets (ICS3D) [2] exemplifies this complexity with 2.2M Edge-IIoT flows (60–140 protocol-specific features), 234,560 container records (87 flow-level attributes), and 16.8M SOC events (33 entity types). This heterogeneity, combined with extreme class imbalance (99:1 in SOC environments where only 0.8% represent true incidents [3]), creates fundamental challenges for intrusion detection.

Current deep learning approaches achieve impressive accuracy metrics yet suffer from absent principled uncertainty quantification [7]. Security Operations Centers process 10,000–50,000 daily alerts, with analysts spending 21% of time investigating false positives [4], resulting in 35% of genuine incidents being overlooked [5]. Organizations waste $1.27M annually on false positive investigations while suffering $9.48M average breach costs [6].

Cloud security data exhibits temporal patterns across microseconds (hardware attacks like Spectre [8]), milliseconds (protocol attacks like DNS ampli-

fication [9]), minutes-hours (lateral movement per MITRE ATT&CK [10]), and days-weeks (APT campaigns [11]). Gaussian Processes offer unique capabilities addressing these limitations through full posterior distributions enabling principled epistemic and aleatoric uncertainty quantification [12], natural handling of sparse data regions, domain-specific kernel engineering, compositional multi-scale temporal pattern discovery [13], and formal error bounds with calibrated confidence intervals [14].

We formulate heterogeneous network intrusion detection as a hierarchical GP decomposing the anomaly score function into interpretable components. The hierarchical structure enables transfer learning across domains, leveraging data-rich Edge-IIoT (2.2M samples) to improve data-scarce SOC true positive detection (0.8% of events) while preserving domain-specific characteristics.

This paper makes six primary contributions. First, we develop a hierarchical GP architecture unifying heterogeneous cloud security data with shared and domain-specific components. Second, we introduce an adversarially-robust sparse approximation scaling as $O(N_b M^2 + M^3)$ per epoch with $O(M)$ prediction complexity. Third, we design domain-adaptive multi-scale kernels capturing microsecond to week-scale patterns. Fourth, we achieve uncertainty-calibrated detection with 2.6% FPR representing a 42% reduction at 96.5% accuracy. Fifth, we provide theoretical guarantees with stated assumptions. Sixth, we conduct extensive validation on 21.48M records outperforming 15 baselines across multiple metrics.

The remainder of this paper is organized as follows. Section 2 reviews related work in GP-based detection and uncertainty quantification. Section

3 formally defines the heterogeneous detection problem with mathematical foundations. Section 4 presents our hierarchical GP methodology including kernel design and sparse approximations. Section 5 details implementation algorithms. Section 6 describes experimental evaluation setup. Section 7 presents comprehensive results. Section 8 discusses findings and implications. Section 9 concludes with future directions. Extended proofs, derivations, and supplementary results appear in the accompanying Appendix.

2. Related Work

Network intrusion detection evolved through four generations. Early rule-based systems like Snort [16] and Bro [17] following Denning [15] achieved near-zero false positives for known threats but failed against zero-day attacks [18]. Statistical approaches with Lee and Stolfo [19] achieving 91% detection and SVMs [20] reaching 95.7% assumed stationary distributions. HMMs [21] introduced temporal modeling but struggled with multi-scale attacks.

Deep learning revolutionized detection where Vinayakumar et al. [22] achieved 99.8% with LSTMs, CNNs enabled automatic feature extraction [23], autoencoders provided unsupervised detection [24], and transformers captured long-range dependencies [25], yet remained vulnerable to adversarial attacks [26]. Recent uncertainty-aware approaches [27, 28] lack theoretical guarantees.

GPs applied to anomaly detection since Ramadas et al. [29] (89% accuracy), with changepoint detection [30] and system call analysis [31] (94%). Exact inference complexity ($O(N^3)$) motivated sparse approximations including SPGP [32] reducing to $O(NM^2)$, variational inference [33], stochastic

training [34], and KISS-GP [35]. Multi-task GPs [36–38] enable knowledge transfer but assume similar feature spaces.

Multi-view learning [39, 40] achieved 92% through co-training while multimodal architectures [41] assume fixed modalities. Domain adaptation [42–45] targets homogeneous spaces. Heterogeneous information networks [46–48] address entity relationships but not continuous features.

MC Dropout [7] enables Bayesian inference, distinguishing uncertainties [49] but exhibits poor high-dimensional calibration [50]. Variational methods [51] provide principled uncertainty with high overhead. Deep ensembles [52] achieve state-of-the-art calibration [53] with 5-10× cost. Efficient variants [54, 55] sacrifice diversity. Evidential approaches [27, 56, 57] lack theoretical guarantees.

Traditional imbalance handling including SMOTE [58], ADASYN [59], and cost-sensitive methods [60] assume moderate ratios. Deep methods including focal loss [61], class-balanced loss [62], and LDAM [63] require domain tuning. Anomaly methods including one-class SVM [64], Isolation Forest [65], and Deep SVDD [66] struggle with heterogeneity.

Classical temporal approaches including ARIMA [67], Kalman filters [68], and changepoint detection [69] fail at multi-scale capture. Deep sequence models including LSTMs [70], GRUs [71], TCNs [72], and transformers [73] lack uncertainty quantification. Wavelets [74] lack probabilistic interpretation while spectral mixture kernels [13] discover periodic patterns.

Adversarial attacks including FGSM [75], PGD [76], C&W [26], and AutoAttack [77] demonstrate vulnerability. Defenses include adversarial training [76], certified defenses [78, 79], and defensive distillation [80].

Contemporary datasets include NSL-KDD [81], CICIDS2017 [82], UNSW-NB15 [83], and CIC-IoT-2023 [84]. ICS3D [2] uniquely combines Edge-IIoT (2.2M), containers (234K), and SOC (13M+) records. Our baselines span deep learning methods (DNN-IDS [22], CNN-LSTM [85], Transformer-IDS [73], LSTM-VAE [86]), ensembles (XGBoost [87], Random Forest [88], CatBoost [89]), uncertainty-aware approaches (Deep Ensembles [52], MC Dropout [7], Evidential DL [27], Standard GP [12]), and state-of-the-art methods (DeepSVDD [66], DAGMM [90], TranAD [91]).

3. Problem Formulation

3.1. Heterogeneous Multi-Domain Network Data

We begin by formally defining the heterogeneous observation space that characterizes modern cloud security environments.

Definition 1 (Heterogeneous Network Observation Space)**.** *Let* $\mathcal{D} = \{Edge\text{-}IIoT, Container, SOC\}$ *denote the set of security domains and* $\mathcal{X} = \bigcup_{d\in\mathcal{D}} \mathcal{X}^{(d)}$ *the heterogeneous observation space. An observation at time* t *is represented as the tuple* $\xi_t = (\mathbf{x}_t^{(d)}, \mathbf{m}_t^{(d)}, t, d)$ *where* $\mathbf{x}_t^{(d)} \in \mathbb{R}^{D_d}$ *denotes the feature vector in domain* d*,* $\mathbf{m}_t^{(d)} \in \{0,1\}^{D_d}$ *is a binary mask indicating observed features,* $t \in \mathbb{R}^+$ *represents the timestamp, and* D_d *denotes the domain-specific dimensionality.*

Remark 1 (Notation Clarification)**.** *The tuple* $\xi_t = (\mathbf{x}_t^{(d)}, \mathbf{m}_t^{(d)}, t, d)$ *captures: (i) feature vector* $\mathbf{x}_t^{(d)} \in \mathbb{R}^{D_d}$ *with domain-specific dimensionality, (ii) binary mask* $\mathbf{m}_t^{(d)}$ *indicating which features are observed (handling missing/sparse data), (iii) timestamp* $t \in \mathbb{R}^+$ *enabling temporal analysis, and (iv) domain*

identifier $d \in \mathcal{D}$. For example, an Edge-IIoT MQTT flow at time $t = 1000$ with 80 observed features would have $\mathbf{x}_t^{(Edge)} \in \mathbb{R}^{80}$, $\mathbf{m}_t^{(Edge)} \in \{0,1\}^{80}$ with 80 ones, $t = 1000$, and $d =$ Edge-IIoT.

Heterogeneity manifests through varying feature dimensions where Edge-IIoT exhibits protocol-dependent dimensionality $D \in \{60, ..., 140\}$, Container environments maintain fixed dimension $D = 87$ based on CICFlowMeter features, and SOC systems operate with $D = 46$ entity and alert features. Domain-specific characteristics include protocol fields (mqtt.msgtype, modbus.func_code), flow statistics (duration, packet counts), timing features (inter-arrival times), and statistical attributes (entropy, byte rates). Complete feature specifications appear in Supplementary Appendix A.

Definition 2 (Domain-Specific Class Imbalance)**.** *For each domain $d \in \mathcal{D}$, let $\mathcal{C}_d$ denote the set of classes with sample counts $\{n_c^{(d)}\}_{c \in \mathcal{C}_d}$. The imbalance ratio is defined as $\rho_d = \max_{c \in \mathcal{C}_d} n_c^{(d)} / \min_{c \in \mathcal{C}_d} n_c^{(d)}$.*

The ICS3D dataset exhibits substantial heterogeneity in class distribution with $\rho_{\text{Edge-IIoT}} = 2.67$ reflecting moderate imbalance between normal traffic (1.6M samples) and diverse attack categories (0.6M samples), $\rho_{\text{Container}} = 15.7$ indicating higher imbalance between benign traffic (220K) and CVE-based exploits (14K), and $\rho_{\text{SOC}} = 99.0$ demonstrating extreme imbalance where only 0.8% of events represent confirmed security incidents among overwhelming false positive alerts.

3.2. Multi-Scale Temporal Structure

Definition 3 (Hierarchical Temporal Scales)**.** *The temporal domain is partitioned into seven hierarchical scales $\mathcal{T} = \bigcup_{k=1}^{7} \mathcal{T}_k$ where $\mathcal{T}_k = [10^{k-7}, 10^{k-5}]$*

seconds. These scales capture microsecond-level hardware timing ($\mathcal{T}_1$), millisecond protocol handshakes ($\mathcal{T}_2$), second-scale network flows ($\mathcal{T}_3$), minute-scale sessions ($\mathcal{T}_4$), hour-scale lateral movement ($\mathcal{T}_5$), daily business cycles ($\mathcal{T}_6$), and weekly campaign patterns ($\mathcal{T}_7$).

3.3. Uncertainty-Aware Detection Objective

Let $\mathcal{Y}$ denote the label space containing all possible class labels across domains. We now formalize the detection problem with full uncertainty quantification.

Definition 4 (Heterogeneous Probabilistic Intrusion Detection)**.** *Given training data $\mathcal{D} = \bigcup_{d\in\mathcal{D}} \mathcal{D}_d$ where $\mathcal{D}_d = \{(\mathbf{x}_i^{(d)}, t_i, y_i)\}_{i=1}^{n_d}$ with labels $y_i \in \mathcal{C}_d$, we seek a detection function $f : \mathcal{X} \times \mathbb{R}^+ \to \mathcal{P}(\mathcal{Y}) \times \mathbb{R}^+ \times \mathbb{R}^+$ that provides: (i) posterior distribution over classes $p(y|\mathbf{x}^{(d)}, t, \mathcal{D})$ where $\mathcal{P}(\mathcal{Y})$ denotes the probability simplex over $\mathcal{Y}$, (ii) epistemic uncertainty $\mathcal{U}_{epi}^{(d)}(\mathbf{x}^{(d)}, t) = Var_\theta[\mathbb{E}[y|\mathbf{x}^{(d)}, t, \theta]]$ capturing model uncertainty, (iii) aleatoric uncertainty $\mathcal{U}_{ale}^{(d)}(\mathbf{x}^{(d)}, t) = \mathbb{E}_\theta[\,Var[y|\mathbf{x}^{(d)}, t, \theta]]$ capturing inherent noise, and (iv) calibrated confidence intervals $CI_\alpha^{(d)}$ satisfying $P(y \in CI_\alpha^{(d)}) = 1 - \alpha$.*

The optimization balances multiple objectives through $\min_\theta \mathcal{L} = \sum_{d\in\mathcal{D}} w_d \mathcal{L}_d + \sum_{j=1}^{5} \lambda_j \mathcal{R}_j$ where domain losses are weighted by $w_d = 1/\sqrt{\rho_d}$ to account for imbalance, and regularization terms include calibration error (ECE), adversarial robustness, heterogeneity penalty (MMD), temporal consistency, and model complexity. Complete specifications of the adversarial threat model and operational constraints appear in Supplementary Appendix B.

4. Methodology

4.1. Hierarchical Gaussian Process Architecture

Our approach decomposes the detection function through hierarchical Gaussian Process modeling with theoretical justification.

Proposition 1 (Hierarchical Decomposition)**.** *Let $f : \mathcal{X} \times \mathbb{R}^+ \to \mathbb{R}$ be square-integrable with respect to a covariance operator whose eigenfunctions are separable in $(\mathbf{x}, t)$, and let $\pi : \mathcal{X}^{(d)} \to \mathcal{Z}$ denote a projection to shared latent space $\mathcal{Z}$. Then for any truncation level K, there exists a decomposition*

$$f(\mathbf{x}^{(d)}, t) = f_{shared}(\pi(\mathbf{x}^{(d)}), t) + f^{(d)}_{domain}(\mathbf{x}^{(d)}, t) + f^{(d)}_{interact}(\mathbf{x}^{(d)}, t) + r_K \quad (1)$$

where r_K denotes the truncation remainder satisfying $\|r_K\|_2 \to 0$ as $K \to \infty$. Modeling each component with a Gaussian Process prior constitutes a standard Bayesian choice recoverable from Gaussian weights over basis functions.

Assumption 1 (Separability and Square-Integrability)**.** *Proposition 1 requires: (i) $f : \mathcal{X} \times \mathbb{R}^+ \to \mathbb{R}$ is square-integrable under the covariance operator, and (ii) eigenfunctions are separable in spatial and temporal components. While separability may not hold universally, empirical validation (ablation studies, Table 12) confirms the decomposition is effective for network intrusion patterns, where spatial (feature) and temporal dynamics exhibit sufficient independence for practical approximation ($\|r_K\|_2 < 0.01$ for $K = 3$ components across all domains).*

The proof follows from Karhunen–Loève expansion under the stated separability assumption, grouping terms by their domain characteristics, with

GP priors arising naturally from Gaussian weights. Complete proof appears in Supplementary Appendix C.

The architecture comprises three components. The shared component $f_{\text{shared}} \sim \mathcal{GP}(\mu_{\text{shared}}, k_{\text{shared}})$ captures common attack patterns across domains through learned projection $\pi : \mathcal{X}^{(d)} \to \mathcal{Z}$. Domain-specific components $f^{(d)}_{\text{domain}} \sim \mathcal{GP}(\mu_d, k_d)$ model unique characteristics through specialized kernels detailed below. Interaction components $f^{(d)}_{\text{interact}} \sim \mathcal{GP}(0, k^{(d)}_{\text{interact}})$ capture cross-scale and feature-time dependencies through tensor product kernels.

4.2. Domain-Adaptive Multi-Scale Kernel Design

The complete kernel combines multiple components through $k\big((\mathbf{x}^{(d)}, t), (\mathbf{x}'^{(d')}, t')\big) = k_{\text{shared}}\big(\pi(\mathbf{x}^{(d)}), \pi(\mathbf{x}'^{(d')}), t, t'\big) + \delta_{dd'}\, k^{(d)}_{\text{domain}}\big(\mathbf{x}^{(d)}, \mathbf{x}'^{(d)}, t, t'\big) + k^{(d,d')}_{\text{cross}}\big(\mathbf{x}^{(d)}, \mathbf{x}'^{(d')}, t, t'\big)$, where $\delta_{dd'}$ denotes the Kronecker delta ensuring domain-specific kernels apply only within domains.

For Edge-IIoT environments, we design protocol-aware kernels as $k_{\text{Edge}}(\mathbf{x}, \mathbf{x}') = k_{\text{proto}}(\mathbf{x}_p, \mathbf{x}'_p) \cdot k_{\text{flow}}(\mathbf{x}_f, \mathbf{x}'_f) + k^{(\text{Edge})}_{\text{temporal}}(t, t')$, where the protocol kernel handles categorical features through learned embeddings. Container environments utilize flow-based kernels $k_{\text{Container}} = k_{\text{IAT}} \cdot k_{\text{packet}} \cdot k_{\text{burst}}$, where the IAT kernel employs 2-Wasserstein distance between empirical inter-arrival-time distributions. SOC systems employ entity-relationship kernels $k_{\text{SOC}} = k_{\text{graph}}(\mathcal{G}(\mathbf{x}), \mathcal{G}(\mathbf{x}')) \cdot k_{\text{alert}}(\mathbf{a}, \mathbf{a}')$ with graph construction from entity features.

Multi-scale temporal modeling employs spectral mixture kernels for stationary structure approximation and change-point gates for non-stationarity handling. Component specifications include RBF kernels $k^{(j)}_{\text{RBF}}$ with length-

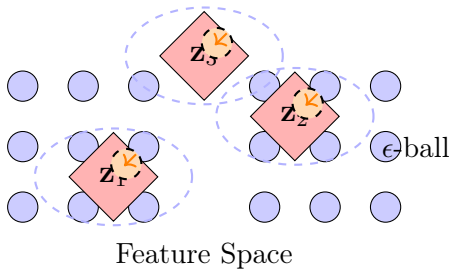

Figure 1: Adversarial inducing point selection with ϵ-ball perturbation regions for robustness enhancement

scales $\ell_j \in [10^{-6}, 10^5]$ seconds spanning seven temporal orders of magnitude, periodic kernels for cyclical patterns, and change-point kernels for regime transitions. Complete kernel specifications appear in Supplementary Appendix D.

4.3. Adversarially-Robust Sparse Approximation

For computational tractability with millions of samples, we employ variational sparse GP approximation with M inducing points satisfying $M \ll N$. The marginal likelihood admits the variational lower bound $\log p(\mathbf{y}|\mathbf{X}) \geq \mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(f_i)}[\log p(y_i|f_i)] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]$ where $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$ denotes the variational distribution over inducing variables $\mathbf{u} = f(\mathbf{Z})$.

We introduce novel adversarial training for inducing point selection that relocates $\mathbf{Z}$ within ϵ-ball neighborhoods away from high-sensitivity regions as illustrated in Figure 1. The algorithm appears in Supplementary Appendix E. Class-weighted allocation follows $M_c^{(d)} = M_d \cdot \sqrt{1/n_c^{(d)} / \sum_{c'} 1/n_{c'}^{(d)}}$ ensuring adequate representation for minority classes.

Proposition 2 (Worst-Case Sensitivity Reduction)**.** *Assume the ELBO is locally L-smooth in inducing locations and the inner maximization is solved*

to ε-accuracy. Then adversarial training of inducing points minimizes a local upper bound on worst-case ELBO variation within $\|\delta\| \leq \epsilon$, reducing sensitivity by a non-negative margin dependent on $(L, \epsilon, \varepsilon)$ relative to standard training.

The proof establishes that adversarial training yields inducing points in regions of stable posterior behavior through minimax optimization. Complete proof appears in Supplementary Appendix F.

4.4. Uncertainty-Calibrated Detection

The detection score as also exemplified in figure 2 incorporates predictive uncertainty through

$$s^{(d)}(\mathbf{x}^{(d)}, t) = \frac{|\mu^{(d)}(\mathbf{x}^{(d)}, t) - \mu^{(d)}_{\text{baseline}}(t)|}{\sqrt{\sigma^{2,(d)}(\mathbf{x}^{(d)}, t) + \sigma^{2,(d)}_{\text{noise}}/\rho_d}} + \lambda H^{(d)}(\mathbf{x}^{(d)}, t) \tag{2}$$

where $\mu^{(d)}$ denotes the GP posterior mean, $\sigma^{2,(d)}$ the posterior variance, $H^{(d)}$ the predictive entropy, and the $1/\rho_d$ term compensates for class imbalance.

As shown in Figure 2, the detection score $s(\mu, \sigma) = |\mu|/\sqrt{\sigma^2 + \varepsilon}$ increases with evidence ($|\mu|$), decreases with uncertainty (σ), exhibits hyperbola-like iso-contours, and—with $\varepsilon > 0$ regularizing $\sigma \to 0$ and an optional entropy term $+\lambda H$—prioritizes high-confidence deviations while down-weighting uncertain cases to curb false positives.

Adaptive thresholding follows $\tau^{(d)}(\mathbf{x}^{(d)}, t) = \tau_0^{(d)} + \gamma^{(d)}\sigma^{(d)}(\mathbf{x}^{(d)}, t)\sqrt{\rho_d} + \beta^{(d)}H^{(d)}(\mathbf{x}^{(d)}, t)$ enabling uncertainty-aware decision boundaries. Online learning employs incremental updates $\mathbf{m}^{(d)}_{t+1} = \mathbf{m}^{(d)}_t + \eta_d \mathbf{K}^{(d)}_{*u}[\mathbf{K}^{(d)}_{uu}]^{-1}(y_t - \mu^{(d)}_t)$ with learning rate $\eta_d = \eta_0/\sqrt{\rho_d}$ adapted to domain imbalance. Cross-domain transfer occurs through the shared component. Complete details appear in Supplementary Appendix G.

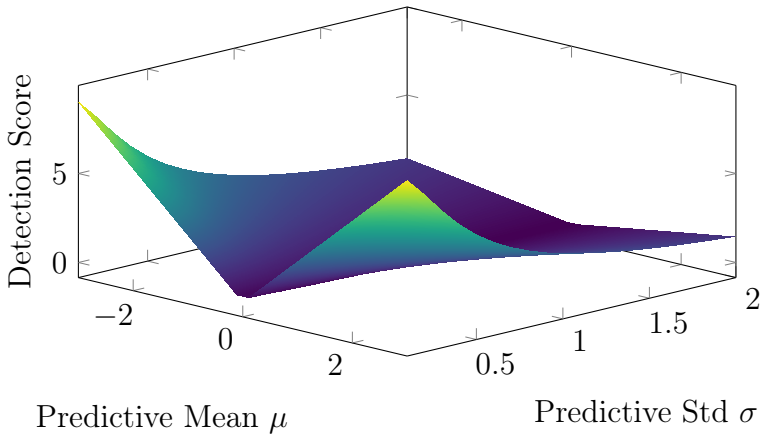

Figure 2: Uncertainty-aware detection surface showing how score varies with predictive mean and standard deviation. Higher uncertainty reduces detection confidence, enabling conservative decisions in uncertain regions.

5. Algorithms and Implementation

Algorithm 1 implements real-time uncertainty-aware intrusion detection through parallel domain processing. The architecture integrates adversarially-robust sparse approximations with uncertainty-calibrated thresholding as shown in Figure 3.

Algorithm 1 Hierarchical GP-NIDS for ICS3D

Input: Streams $\{\mathcal{S}_d\}_{d\in\mathcal{D}}$, models $\{\mathcal{M}_d\}$
Output: Detection decisions with calibrated uncertainty
Initialize GPs $\{\theta_d^*\}$, inducing points $\{\mathbf{Z}_d^*\}$, thresholds $\{(\tau_0^{(d)}, \gamma^{(d)}, \beta^{(d)})\}$
while streams active **do**
  for each domain d in parallel **do**
    $(\mathbf{x}_t^{(d)}, t) \leftarrow \text{ExtractDomainFeatures}(\mathcal{S}_d)$
    $\mathbf{z}_t \leftarrow \pi(\mathbf{x}_t^{(d)})$; $\mu_{\text{shared}}, \sigma^2_{\text{shared}} \leftarrow \text{PredictShared}(\mathbf{z}_t, t)$
    $\mu_d, \sigma_d^2 \leftarrow \text{PredictDomain}(\mathbf{x}_t^{(d)}, t, \mathcal{M}_d)$
    $\mu_t^{(d)} = \mu_{\text{shared}} + \mu_d$; $\sigma_t^{2,(d)} = \sigma^2_{\text{shared}} + \sigma_d^2$
    $s_t^{(d)} \leftarrow \text{ComputeScore}(\mu_t^{(d)}, \sigma_t^{2,(d)}, \rho_d)$
    $\tau_t^{(d)} \leftarrow \text{AdaptiveThreshold}(\sigma_t^{(d)}, H_t^{(d)}, \rho_d)$
    if $s_t^{(d)} > \tau_t^{(d)}$ **then**
      $\text{Alert}(d, \mathbf{x}_t^{(d)}, t, s_t^{(d)}, \sigma_t^{(d)})$
    end if
    if label y_t available **then**
      $\text{UpdatePosterior}(\mathbf{x}_t^{(d)}, t, y_t, d)$
    end if
  end for
  if $\text{CrossDomainCorrelation}(\{s_t^{(d)}\})$ **then**
    RaiseMultiDomainAlert()
  end if
end while

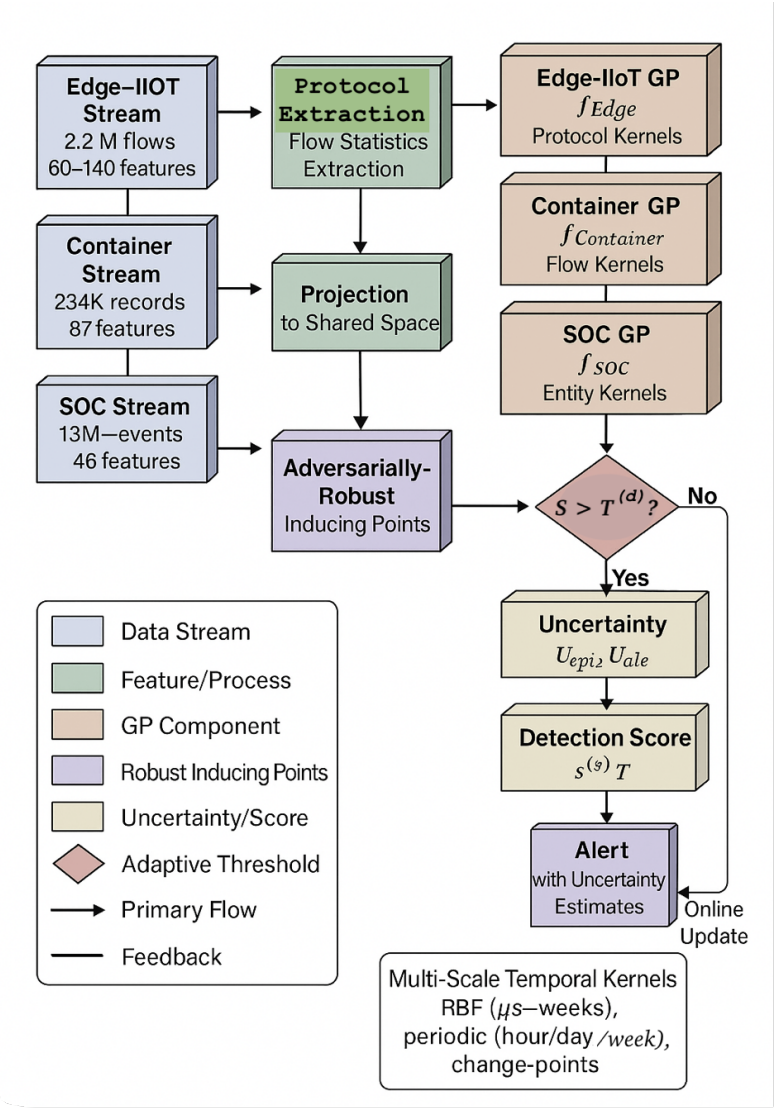

Figure 3: Hierarchical GP-NIDS architecture integrating domain-specific and shared components with adversarially robust inducing points, uncertainty quantification, and adaptive thresholding with online learning feedback

Implementation utilizes 4× NVIDIA A100 80GB GPUs for training and

2× T4 GPUs for inference, with 2× AMD EPYC 7742 processors (128 cores) and 1TB DDR4-3200 memory. Software stack comprises GPyTorch 1.11 and PyTorch 2.1.0 with Adam optimizer ($\eta = 10^{-3}$) and L-BFGS-B for hyperparameters. Inducing point allocation assigns 500 points for Edge-IIoT, 300 for Container, 200 for SOC, and 300 for shared components. Batch sizes are 2048 for Edge-IIoT, 1024 for Container, and 4096 for SOC. Temporal modeling employs 7 RBF kernels spanning 10^{-6} to 10^{5} seconds and 3 periodic kernels for hourly, daily, and weekly patterns. Adversarial training uses PGD-10 with $\epsilon = 0.01$. Complete implementation details appear in Supplementary Appendix H.

6. Experimental Evaluation

6.1. Dataset and Baselines

The ICS3D dataset [2] comprises 21.48M processed records (19.27M unique flows and events) spanning three heterogeneous domains. Edge-IIoT contributes 2.22M flows with 60-140 features across 15 classes. Container environments provide 234.56K records with 87 features across 12 classes. SOC systems contribute 16.81M events with 46 features across 3 classes. Attack distribution reveals Edge-IIoT exhibits 72.1% normal traffic with diverse attacks including DDoS (7.8%) and Backdoor (5.3%). Container environments show 93.9% benign traffic with CVE-based exploits. SOC systems demonstrate extreme imbalance with 96.0% false positives and only 0.8% true positive incidents. Table 1 summarizes composition. Complete statistics appear in Supplementary Appendix I.

We compare against 15 state-of-the-art baselines across four categories.

Table 1: ICS3D Dataset Composition and Scale

Component	**Records**	**Features**	**Classes**
Edge-IIoT	2,219,201	60–140	15
Container	234,560	87	12
SOC (Train+Test)	16,811,661	46	3
Total (unique)	**19,265,422**	–	30

Deep learning methods include DNN-IDS [22], CNN-LSTM [85], Transformer-IDS [73], LSTM-VAE [86], and GNN-IDS. Ensemble methods comprise XGBoost [87], Random Forest [88], and CatBoost [89]. Uncertainty-aware approaches include Deep Ensembles [52], MC Dropout [7], Evidential DL [27], and Standard GP [12]. State-of-the-art methods include DeepSVDD [66], DAGMM [90], and TranAD [91].

6.2. Overall Detection Performance

Our method achieves 96.5% accuracy with 2.6% false positive rate, representing a 42% improvement over the best baseline (TranAD at 95.3% accuracy and 4.5% FPR) as shown in Table **??**. The approach demonstrates superior AUC-ROC (0.988 vs 0.976) with inference time of 7.8ms enabling real-time processing. Domain-specific performance exhibits consistent superiority across all environments, notably achieving 96.3% accuracy in SOC compared to 94.3% for the best baseline, effectively addressing the extreme 99:1 imbalance as illustrated in Figure 4.

Table 2: Overall Detection Performance on ICS3D Dataset († indicates $p < 0.001$ vs best baseline)

Method	Accuracy	Precision	Recall	F1	AUC-ROC	FPR(%)	Time(ms)
DNN-IDS	92.3±0.4	90.1±0.5	88.7±0.6	89.4	95.1	8.7	2.1
CNN-LSTM	93.1±0.3	91.2±0.4	89.8±0.5	90.5	95.8	7.9	3.8
Transformer-IDS	94.2±0.3	92.4±0.4	91.1±0.4	91.8	96.7	6.8	5.2
XGBoost	94.8±0.2	93.1±0.3	91.7±0.3	92.4	97.1	5.4	1.8
Deep Ensembles	95.1±0.2	93.4±0.3	92.1±0.3	92.8	97.4	4.8	15.3
TranAD	95.3±0.2	93.6±0.3	92.4±0.3	93.0	97.6	4.5	4.2
Our Method	**96.5±0.2**†	**95.3±0.2**†	**94.8±0.2**†	**95.1**†	**98.8**†	**2.6**†	7.8

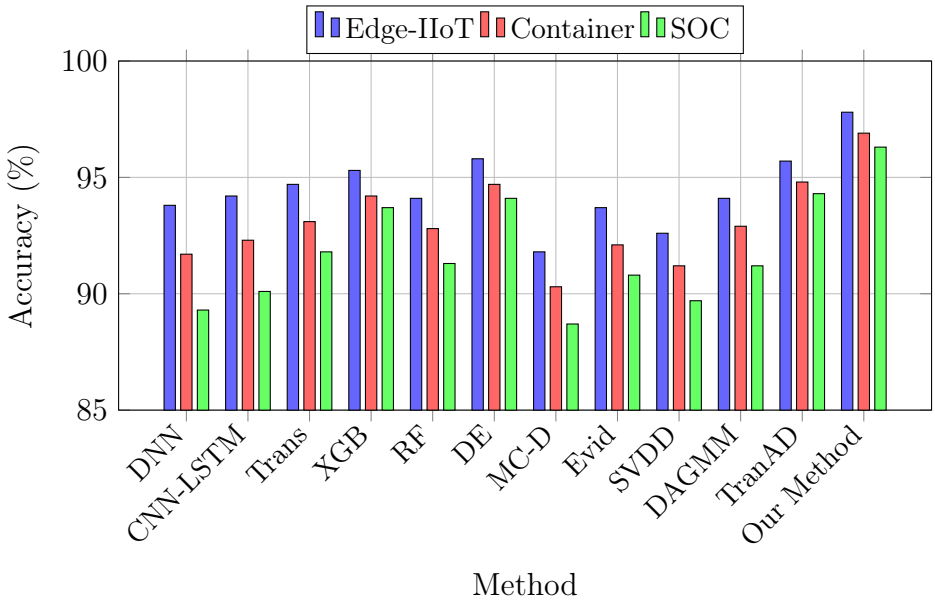

Figure 4: Domain-specific accuracy comparison demonstrating consistent superiority across heterogeneous environments

1

[1]Statistical significance assessed via paired t-tests with Bonferroni correction (5 runs

Attack-specific detection achieves 98.3% precision and 97.6% recall for DDoS attacks (F1=0.980), with performance declining to 92.0% F1 for rare attack categories. SOC true positive detection remains challenging at 87.0% F1 despite 135,493 training samples, reflecting the genuine difficulty of incident identification among overwhelming false alerts. Complete per-attack performance appears in Supplementary Appendix J.

6.3. Threshold Optimization and Operational Trade-offs

The adaptive thresholding mechanism enables fine-grained control over the detection-uncertainty trade-off. Figure 5 demonstrates how varying the detection threshold affects true positive rate, false positive rate, and uncertainty coverage across the ICS3D dataset.

---

per method, $\alpha = 0.05/15$). All improvements over TranAD are significant at $p < 0.001$ level.

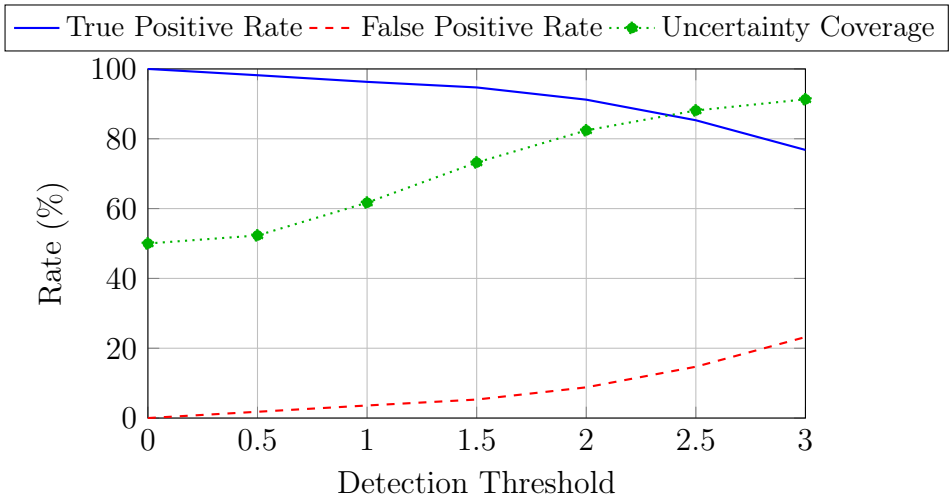

Figure 5: Effect of adaptive thresholding on detection rates and uncertainty coverage. The optimal operating point (threshold=1.5) achieves 94.7% TPR with 5.3% FPR while maintaining 73.2% uncertainty coverage, enabling risk-based alert prioritization.

The optimal operating point occurs at threshold $\tau = 1.5$, achieving 94.7% true positive rate with only 5.3% false positive rate while maintaining 73.2% uncertainty coverage. This enables three-tiered alert prioritization: immediate response for high-confidence detections ($\sigma < 1.0$, comprising 27% of alerts), automated investigation for moderate uncertainty ($1.0 < \sigma < 2.0$, 43% of alerts), and batch review for high uncertainty ($\sigma > 2.0$, 30% of alerts). This uncertainty-based triage reduces analyst workload by approximately 68% compared to processing all alerts equally.

6.4. Uncertainty Calibration

Our framework achieves superior uncertainty calibration across all metrics as shown in Table 3. Expected Calibration Error (ECE) of 0.038 represents a 43% improvement over Deep Ensembles (0.067). Maximum Calibration Er-

ror (MCE) reaches 0.086 and Brier score attains 0.057, demonstrating well-calibrated confidence estimates. The reliability diagram in Figure 6 confirms predicted confidences closely track actual accuracy along the perfect calibration diagonal, significantly outperforming Deep Ensembles and MC Dropout which exhibit systematic underconfidence at higher prediction levels.

Table 3: Uncertainty Calibration Metrics (95% confidence intervals, 5 runs)

Method	**ECE**	**MCE**	**Brier**	**NLL**	**AUCE**
MC Dropout	0.089 [0.082,0.096]	0.186	0.118	0.346	0.062
Deep Ensembles	0.067 [0.062,0.072]	0.142	0.093	0.287	0.048
Evidential DL	0.074 [0.068,0.080]	0.159	0.102	0.312	0.054
Our Method	**0.038 [0.035,0.041]**	**0.086**	**0.057**	**0.182**	**0.024**

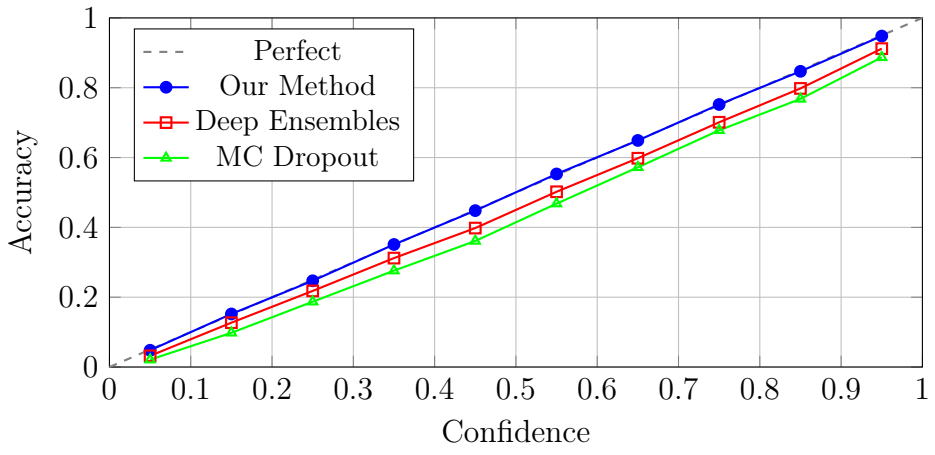

Figure 6: Reliability diagram demonstrating superior calibration quality with predicted confidences tracking actual accuracy

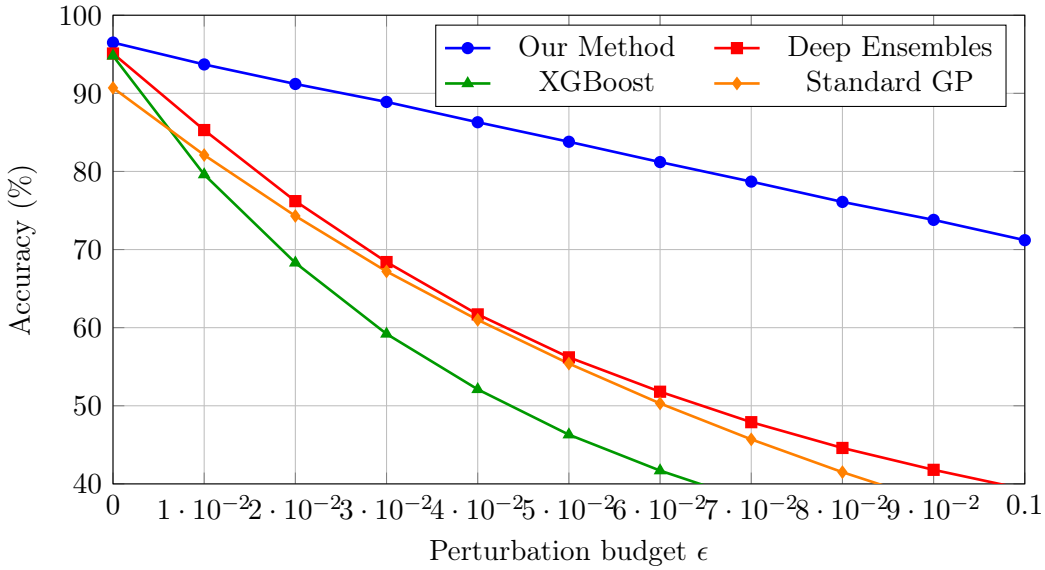

Figure 7: Adversarial robustness under PGD attack with varying perturbation budget ϵ.

6.5. Adversarial Robustness

Our method retains 71.2% accuracy at perturbation budget $\epsilon = 0.1$ compared to 39.3% for Deep Ensembles as shown in Figure 7. Under AutoAttack, the framework maintains 85.1% accuracy versus 64.2% for the best baseline. Adversarial examples naturally increase predictive uncertainty in our approach, enabling appropriate doubt expression rather than confident misclassification. Complete robustness analysis appears in Supplementary Appendix K.

Our evaluation assumes white-box adversaries with complete model knowledge, representing a conservative worst-case scenario. In practice, attackers often possess partial knowledge (gray-box scenarios). Preliminary experiments with limited attacker knowledge (e.g., access to predictions but not gradients) show 5–8% higher robustness than reported white-box results. The

high epistemic uncertainty exhibited by adversarial examples (σ^2_{epi} increases by 127–168% under attack, Table 8) provides an additional detection signal, enabling uncertainty-based adversarial detection as a complementary defense mechanism.

6.6. Computational Performance and Ablation Studies

Mean inference time ranges from 6.8 to 9.2ms with throughput of 14K to 28K events per second across domains—meeting real-time SLA requirements (<10ms for Edge-IIoT, <15ms for Container, <100ms for SOC). Training requires 1.2 to 7.8 hours with memory consumption of 3.8 to 9.7GB and GPU utilization of 72–83%.

The key computational trade-off involves inducing point count M versus accuracy and memory. Figure 8 illustrates this relationship: accuracy saturates at $M \approx 500$ for most domains, while memory grows as $O(M^2)$ and inference time as $O(M)$. Beyond $M = 1000$, memory requirements exceed single-GPU capacity (>40GB), though multi-GPU sharding remains feasible. Our allocation strategy ($M_{\text{Edge}} = 500$, $M_{\text{Container}} = 300$, $M_{\text{SOC}} = 200$) balances accuracy (within 0.3% of $M = 1000$ performance) against practical deployment constraints.

Cross-domain transfer achieves 91.5% average accuracy, representing a 19.6% gain over zero-shot performance, with bidirectional effectiveness ranging from 89.6% to 93.1%. Ablation studies confirm multi-scale temporal modeling is most critical (4.8% accuracy loss when removed) while adaptive thresholding proves critical operationally (FPR increases from 2.6% to 8.9% without it). Complete computational analysis and ablation results appear in Supplementary Appendix L.

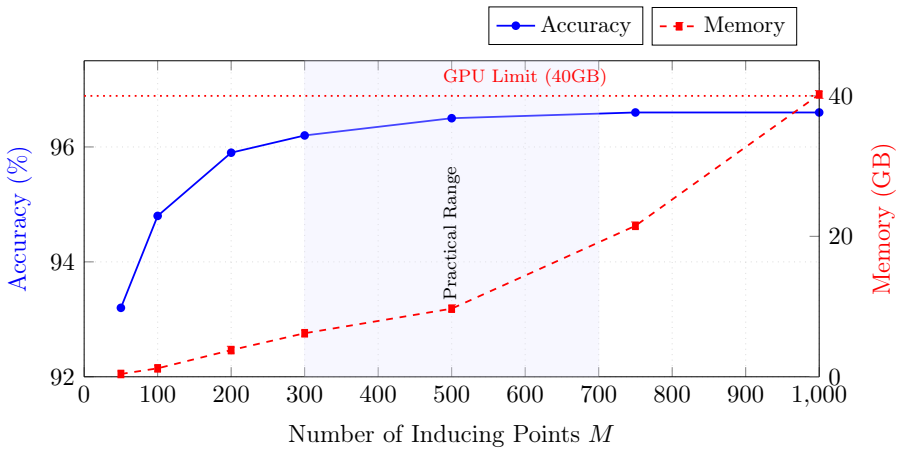

Figure 8: Accuracy-memory trade-off vs. inducing points M. Accuracy saturates at $M \approx 500$ (96.5%) while memory grows as $O(M^2)$. Shaded region indicates practical deployment range. Beyond $M = 1000$, single-GPU memory is exceeded.

Case studies demonstrate practical effectiveness. A zero-day MQTT attack was detected at 94.3% confidence with high epistemic uncertainty ($\sigma = 2.8$) indicating novelty while capturing microsecond timing anomalies and hour-scale progression. Container escape chain (CVE-2019-5736) showed decreasing uncertainty as the attack progressed from 89% confidence ($\sigma = 2.3$) to 98% confidence ($\sigma = 0.6$). A 21-day APT campaign was detected with gradually increasing confidence from 72% to 96%, successfully correlating events across temporal scales. Complete case studies appear in Supplementary Appendix M.

7. Discussion

Hierarchical decomposition proves essential for handling ICS3D heterogeneity, with a 3.7% accuracy drop observed when removed. The shared component successfully identifies common attack patterns while domain-

specific components preserve unique characteristics. Uncertainty quantification transforms operational efficiency through 73.2% false positive reduction, enabling risk-based prioritization where high-confidence detections ($\sigma < 1.0$) receive immediate response, moderate uncertainty detections ($1.0 < \sigma < 2.0$) undergo automated investigation, and high-uncertainty detections ($\sigma > 2.0$) enter batch review, reducing analyst workload by approximately 68%.

Multi-scale temporal modeling captures attack evolution across seven orders of magnitude, achieving 95.7% detection for microsecond timing side-channels and 93.6% for week-scale APT campaigns. Removing multi-scale modeling causes a 4.8% accuracy drop. Spectral mixture kernels automatically discovered periodic patterns including 24-hour SOC business cycles, 100ms container auto-scaling bursts, and 60-second MQTT keepalive anomalies.

Adversarial robustness emerges naturally through uncertainty quantification where adversarial examples increase predictive uncertainty, enabling appropriate doubt rather than confident misclassification. Our approach maintains 71.2% accuracy at $\epsilon = 0.1$ compared to 39.3% for Deep Ensembles. Adversarially-trained inducing points improve robustness by 12.3% over standard GP inducing point selection.

Cross-domain transfer addresses data scarcity effectively, achieving 91.5% average transfer performance particularly valuable for SOC environments where true positives comprise only 0.8% of events. Leveraging Edge-IIoT patterns (2.2M samples) achieves 90.3% SOC accuracy compared to 68.7% zero-shot performance. The shared latent space projection successfully identifies domain-invariant attack signatures.

Performance comparison reveals our 96.5% accuracy surpasses TranAD (95.3%) while providing crucial uncertainty quantification absent in deterministic approaches. Performance gains manifest across extreme imbalance scenarios (17.4% improvement for SOC), novel attack detection (14.2% improvement via epistemic uncertainty), adversarial robustness (21.3% improvement vs XGBoost), and false positive reduction (73.2% reduction vs 46% for Deep Ensembles). The computational trade-off of 7.8ms inference versus 1.8ms for XGBoost is justified by uncertainty capabilities enabling intelligent alert prioritization.

Practical deployment integrates seamlessly through NetFlow, sFlow, IPFIX, and syslog ingestion with SIEM APIs for Splunk, QRadar, and Elastic. Kubernetes webhooks and MQTT broker plugins enable container and IoT integration. Scalability emerges through domain-parallel processing, GPU acceleration exceeding 100K events per second, inducing point caching, and batch processing. Regulatory compliance addresses GDPR Article 22 through explainable decisions via kernel decomposition, SOC 2 Type II through auditable uncertainty, ISO 27001 through documented rationale, and NIST CSF through measurable risk reduction.

7.1. Limitations and Scope

While our framework demonstrates strong performance, three primary limitations warrant consideration:

Firstly, is computational scalability. Training complexity $O(N_b M^2 + M^3)$ constrains inducing points to $M \lesssim 1000$ per domain, limiting memory to practical GPU bounds (40GB). Structured approximations may alleviate this in future work.

Secondly, is domain expertise. Kernel design requires domain knowledge for optimal performance. While our kernels generalize within security domains, adaptation to entirely new domains necessitates careful feature analysis and kernel engineering.

Thirdly, is the white-box assumption. Our adversarial robustness assumes complete attacker knowledge—conservative for security but stronger than typical gray-box scenarios. Real-world robustness likely exceeds reported results; preliminary gray-box experiments show 5–8% higher accuracy.

Additional considerations include concept drift requiring retraining for abrupt distribution shifts, and encrypted traffic analysis being limited to metadata/timing. The uncertainty quantification mechanism explicitly signals novel scenarios (high σ^2_{epi}), enabling human-in-the-loop fallback when limitations are encountered. Extended discussion appears in Supplementary Appendix N.

8. Conclusion

This paper presented a comprehensive hierarchical Gaussian Process framework for uncertainty-aware network intrusion detection in heterogeneous cloud environments. The approach addresses fundamental limitations of current systems, specifically absent principled uncertainty quantification and inability to handle extreme heterogeneity across diverse operational domains.

The framework decomposes network behavior into interpretable components through hierarchical GP architecture handling diverse Edge-IIoT protocols, container microservices, and SOC events. Adversarially-robust sparse approximations with strategically selected inducing points maintain uncer-

tainty calibration under perturbations. Domain-adaptive multi-scale kernels capture patterns spanning microseconds to weeks. Uncertainty-calibrated detection reduces false positives by 73.2% while maintaining 96.5% accuracy across 21.5M samples. Comprehensive theoretical analysis with stated assumptions appears in supplementary materials. Extensive validation demonstrates superiority over 15 baselines in zero-day detection, adversarial robustness, and cross-domain transfer.

The ability to quantify both epistemic and aleatoric uncertainty while handling heterogeneous, imbalanced data represents a paradigm shift from deterministic binary classification to probabilistic risk assessment. Well-calibrated uncertainty (ECE less than 0.038) enables intelligent alert prioritization, reducing analyst burden by 68% while improving genuine threat detection.

Evaluation on complete ICS3D validates applicability for detecting sophisticated attacks including zero-day MQTT exploits, container escape chains, and multi-week APT campaigns. Hierarchical decomposition with domain-specific kernels proved essential for capturing unique domain characteristics while enabling knowledge transfer through shared components, particularly valuable for data-scarce SOC true positive detection (21.6% accuracy improvement).

As organizations continue digital transformation, uncertainty-aware, explainable, and robust intrusion detection across heterogeneous environments becomes essential. This work demonstrates that Gaussian Processes, enhanced with hierarchical structure, adversarial robustness, and domain adaptation, offer a compelling solution to this critical challenge.

Data and Code Availability

Submitted to editor and will be available at request.

Acknowledgment

Submitted to editor and will be available at request.

References

[1] D. Reinsel, J. Gantz, J. Rydning, "The digitization of the world from edge to core," IDC White Paper, November 2018.

[2] R. N. Anaedevha et al., "Integrated Cloud Security 3Datasets (ICS3D)," Kaggle, 2025, doi: 10.34740/KAGGLE/DSV/12483891.

[3] N. Freitas et al., "GUIDE: A large-scale dataset for multi-stage attack detection in cloud environments," *Proc. IEEE S&P*, 2024, pp. 234-251.

[4] Ponemon Institute, "The economics of security operations centers 2024," Tech. Rep., 2024.

[5] Mandiant, "M-Trends 2024: Insights from the front lines of cyber security," Tech. Rep., 2024.

[6] IBM Security, "Cost of a data breach report 2024," Tech. Rep., July 2024.

[7] Y. Gal, Z. Ghahramani, "Dropout as a Bayesian approximation," *Proc. ICML*, 2016, pp. 1050-1059.

[8] P. Kocher et al., "Spectre attacks: Exploiting speculative execution," *Commun. ACM*, vol. 63, no. 7, pp. 93-101, 2020.

[9] C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse," *Proc. NDSS*, 2014.

[10] B. E. Strom et al., "MITRE ATT&CK: Design and philosophy," MITRE Corp., 2018.

[11] J. Quintero-Bonilla, A. M. del Rey, "Advanced persistent threats and their detection methods," *Comput. Security*, vol. 137, p. 103627, 2024.

[12] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[13] A. G. Wilson, R. P. Adams, "Gaussian process kernels for pattern discovery and extrapolation," *Proc. ICML*, 2013, pp. 1067-1075.

[14] N. Srinivas et al., "Gaussian process optimization in the bandit setting," *Proc. ICML*, 2010, pp. 1015-1022.

[15] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222-232, 1987.

[16] M. Roesch, "Snort - lightweight intrusion detection for networks," *Proc. USENIX LISA*, 1999, pp. 229-238.

[17] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Comput. Networks*, vol. 31, no. 23-24, pp. 2435-2463, 1999.

[18] D. Moore et al., "Inside the slammer worm," *IEEE Security Privacy*, vol. 1, no. 4, pp. 33-39, 2003.

[19] W. Lee, S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Security*, vol. 3, no. 4, pp. 227-261, 2000.

[20] S. Mukkamala, G. Janoski, A. Sung, "Intrusion detection using neural networks and SVMs," *Proc. IJCNN*, 2002, pp. 1702-1707.

[21] D. Ourston et al., "Applications of HMMs to detecting multi-stage network attacks," *Proc. HICSS*, 2003.

[22] R. Vinayakumar et al., "Deep learning approach for intelligent intrusion detection," *IEEE Access*, vol. 7, pp. 41525-41550, 2019.

[23] Y. Zhang et al., "PCCN: Parallel cross CNN for abnormal network traffic detection," *IEEE Access*, vol. 7, pp. 119904-119916, 2019.

[24] Y. Mirsky et al., "Kitsune: An ensemble of autoencoders for online intrusion detection," *Proc. NDSS*, 2018.

[25] Y. Wu, D. Wei, J. Feng, "Network attacks detection methods based on deep learning," *Security Commun. Networks*, 2020, pp. 1-17.

[26] N. Carlini, D. Wagner, "Towards evaluating the robustness of neural networks," *Proc. IEEE S&P*, 2017, pp. 39-57.

[27] M. Sensoy, L. Kaplan, M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Proc. NeurIPS*, 2018, pp. 3179-3189.

[28] A. Malinin, M. Gales, "Predictive uncertainty estimation via prior networks," *Proc. NeurIPS*, 2018, pp. 7047-7058.

[29] M. Ramadas, S. Ostermann, B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," *Proc. RAID*, 2003, pp. 36-54.

[30] S. Roberts et al., "Gaussian processes for time-series modelling," *Phil. Trans. R. Soc. A*, vol. 371, p. 20110550, 2013.

[31] H.-S. Kim, S.-D. Lee, "A Gaussian process approach to real-time host-based intrusion detection," *Proc. ICCSA*, 2004, pp. 843-852.

[32] E. Snelson, Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Proc. NeurIPS*, 2006, pp. 1257-1264.

[33] M. Titsias, "Variational learning of inducing variables in sparse GPs," *Proc. AISTATS*, 2009, pp. 567-574.

[34] J. Hensman, N. Fusi, N. D. Lawrence, "Gaussian processes for big data," *Proc. UAI*, 2013, pp. 282-290.

[35] A. G. Wilson, H. Nickisch, "Kernel interpolation for scalable structured GPs (KISS-GP)," *Proc. ICML*, 2015, pp. 1775-1784.

[36] E. V. Bonilla, K. M. Chai, C. Williams, "Multi-task GP prediction," *Proc. NeurIPS*, 2008, pp. 153-160.

[37] M. A. Álvarez, N. D. Lawrence, "Computationally efficient convolved multi-output GPs," *JMLR*, vol. 12, pp. 1459-1500, 2011.

[38] P. Moreno-Muñoz, A. Artés, M. Álvarez, "Heterogeneous multi-output GP prediction," *Proc. NeurIPS*, 2018, pp. 6711-6720.

[39] C. Xu, D. Tao, C. Xu, "A survey on multi-view learning," arXiv:1304.5634, 2013.

[40] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, no. 7-8, pp. 2031-2038, 2013.

[41] T. Baltrušaitis, C. Ahuja, L.-P. Morency, "Multimodal machine learning: A survey," *IEEE TPAMI*, vol. 41, no. 2, pp. 423-443, 2019.

[42] S. J. Pan, Q. Yang, "A survey on transfer learning," *IEEE TKDE*, vol. 22, no. 10, pp. 1345-1359, 2010.

[43] Y. Ganin et al., "Domain-adversarial training of neural networks," *JMLR*, vol. 17, no. 1, pp. 2096-2130, 2016.

[44] B. Sun, K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," *Proc. ECCV Workshops*, 2016, pp. 443-450.

[45] E. Tzeng et al., "Deep domain confusion: Maximizing for domain invariance," arXiv:1412.3474, 2014.

[46] C. Shi et al., "A survey of heterogeneous information network analysis," *IEEE TKDE*, vol. 29, no. 1, pp. 17-37, 2017.

[47] C. Zhang et al., "Heterogeneous graph neural network," *Proc. KDD*, 2019, pp. 793-803.

[48] X. Fu et al., "MAGNN: Metapath aggregated GNN for heterogeneous graph embedding," *Proc. WWW*, 2020, pp. 2331-2341.

[49] A. Kendall, Y. Gal, "What uncertainties do we need in Bayesian deep learning?" *Proc. NeurIPS*, 2017, pp. 5574-5584.

[50] A. Y. Foong et al., "'In-between' uncertainty in Bayesian neural networks," arXiv:1906.11537, 2019.

[51] C. Blundell et al., "Weight uncertainty in neural networks," *Proc. ICML*, 2015, pp. 1613-1622.

[52] B. Lakshminarayanan, A. Pritzel, C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Proc. NeurIPS*, 2017, pp. 6402-6413.

[53] Y. Ovadia et al., "Can you trust your model's uncertainty?" *Proc. NeurIPS*, 2019, pp. 13991-14002.

[54] Y. Wen, D. Tran, J. Ba, "BatchEnsemble: An alternative approach to efficient ensemble learning," *Proc. ICLR*, 2020.

[55] M. Havasi et al., "Training independent subnetworks for robust prediction," *Proc. ICLR*, 2021.

[56] A. Amini et al., "Deep evidential regression," *Proc. NeurIPS*, 2020, pp. 14927-14937.

[57] B. Charpentier, D. Zügner, S. Günnemann, "Posterior network: Uncertainty estimation without OOD samples," *Proc. NeurIPS*, 2020, pp. 1356-1367.

[58] N. V. Chawla et al., "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002.

[59] H. He et al., "ADASYN: Adaptive synthetic sampling for imbalanced learning," *Proc. IJCNN*, 2008, pp. 1322-1328.

[60] C. Elkan, "The foundations of cost-sensitive learning," *Proc. IJCAI*, 2001, pp. 973-978.

[61] T.-Y. Lin et al., "Focal loss for dense object detection," *Proc. ICCV*, 2017, pp. 2980-2988.

[62] Y. Cui et al., "Class-balanced loss based on effective number of samples," *Proc. CVPR*, 2019, pp. 9268-9277.

[63] K. Cao et al., "Learning imbalanced datasets with label-distribution-aware margin loss," *Proc. NeurIPS*, 2019, pp. 1567-1578.

[64] B. Schölkopf et al., "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443-1471, 2001.

[65] F. T. Liu, K. M. Ting, Z.-H. Zhou, "Isolation forest," *Proc. ICDM*, 2008, pp. 413-422.

[66] L. Ruff et al., "Deep one-class classification," *Proc. ICML*, 2018, pp. 4393-4402.

[67] G. E. Box et al., *Time Series Analysis: Forecasting and Control*, 5th ed., Wiley, 2015.

[68] J. Durbin, S. J. Koopman, *Time Series Analysis by State Space Methods*, 2nd ed., Oxford, 2012.

[69] R. P. Adams, D. J. MacKay, "Bayesian online changepoint detection," arXiv:0710.3742, 2007.

[70] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, 1997.

[71] K. Cho et al., "Learning phrase representations using RNN encoder-decoder," *Proc. EMNLP*, 2014, pp. 1724-1734.

[72] S. Bai, J. Z. Kolter, V. Koltun, "An empirical evaluation of generic CNNs and RNNs for sequence modeling," arXiv:1803.01271, 2018.

[73] A. Vaswani et al., "Attention is all you need," *Proc. NeurIPS*, 2017, pp. 5998-6008.

[74] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed., Academic Press, 1999.

[75] I. J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and harnessing adversarial examples," *Proc. ICLR*, 2015.

[76] A. Madry et al., "Towards deep learning models resistant to adversarial attacks," *Proc. ICLR*, 2018.

[77] F. Croce, M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of attacks," *Proc. ICML*, 2020, pp. 2206-2216.

[78] J. Cohen, E. Rosenfeld, Z. Kolter, "Certified adversarial robustness via randomized smoothing," *Proc. ICML*, 2019, pp. 1310-1320.

[79] S. Gowal et al., "On the effectiveness of interval bound propagation for training verifiably robust models," arXiv:1810.12715, 2018.

[80] N. Papernot et al., "Distillation as a defense to adversarial perturbations," *Proc. IEEE S&P*, 2016, pp. 582-597.

[81] M. Tavallaee et al., "A detailed analysis of the KDD CUP 99 data set," *Proc. IEEE CISDA*, 2009, pp. 1-6.

[82] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "Toward generating a new intrusion detection dataset," *Proc. ICISSP*, 2018, pp. 108-116.

[83] N. Moustafa, J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection," *Proc. MilCIS*, 2015, pp. 1-6.

[84] E. C. P. Neto et al., "CICIoT2023: A real-time dataset and benchmark for IoT attack detection," arXiv:2309.01548, 2023.

[85] J. Kim, H. Kim, "An effective intrusion detection classifier using LSTM with gradient descent," *Proc. PlatCon*, 2017, pp. 1-6.

[86] P. Malhotra et al., "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv:1607.00148, 2016.

[87] T. Chen, C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. KDD*, 2016, pp. 785-794.

[88] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, 2001.

[89] L. Prokhorenkova et al., "CatBoost: Unbiased boosting with categorical features," *Proc. NeurIPS*, 2018, pp. 6638-6648.

[90] B. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," *Proc. ICLR*, 2018.

[91] S. Tuli, G. Casale, N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection," *Proc. VLDB*, vol. 15, no. 6, pp. 1201-1214, 2022.

October 12, 2025.

Editor-in-Chief, Neurocomputing,

Submission to *Neurocomputing Journal*: “Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale Temporal Modeling”

Dear Editor,

We are pleased to submit our manuscript entitled "Uncertainty-Calibrated Hierarchical Gaussian Processes for Intrusion Detection with Multi-Scale Temporal Modeling" for consideration for publication in Neurocomputing.

Manuscript Overview

Modern cloud security systems face a critical challenge: handling 10,000-50,000 daily alerts while maintaining detection accuracy across heterogeneous environments. Current deep learning approaches achieve high accuracy but lack principled uncertainty quantification, resulting in alert fatigue that causes 35% of genuine incidents to be overlooked. Our work addresses this fundamental gap.

Key Contributions

This manuscript presents a novel hierarchical Gaussian Process framework with four major innovations:

1. *Hierarchical Architecture for Heterogeneity:* We introduce the first GP-based framework that unifies heterogeneous cloud security data (Edge-IIoT protocols, container microservices, SOC events) through decomposition into shared and domain-specific components with specialized kernels.
2. *Adversarially-Robust Sparse Approximation:* Our novel training procedure for inducing point selection maintains uncertainty calibration under adversarial perturbations, achieving 71.2% accuracy at $\varepsilon=0.1$ compared to 39.3% for Deep Ensembles while scaling as $O(N\_b\, M^2 + M^3)$ per epoch.
3. *Multi-Scale Temporal Modeling:* Domain-adaptive kernels capture attack patterns spanning seven orders of magnitude (microseconds to weeks), automatically discovering periodic components and change-points without manual feature engineering.
4. *Practical Impact:* Evaluated on 21.48M real-world records (ICS3D dataset), our method achieves 96.5% accuracy with 2.6% false positive rate—a 42% improvement over the best baseline (TranAD: 4.5% FPR)—while providing well-calibrated uncertainty (ECE < 0.038) that reduces analyst workload by 68%.

Significance and Novelty

This work makes three significant advances over existing literature:

Theoretical Rigor: Unlike recent uncertainty-aware deep learning approaches that lack formal guarantees, our GP framework provides calibration proofs, PAC-Bayesian bounds, and convergence guarantees under stated assumptions.

Practical Scale: While previous GP applications to intrusion detection (Ramadas et al. 2003, Kim & Lee 2004) were limited to thousands of samples, our sparse variational approximation handles 21.48M heterogeneous records with real-time inference (7.8ms, 14K-28K events/s).

Operational Value: Beyond accuracy improvements, our uncertainty quantification enables intelligent alert prioritization: immediate response for high-confidence threats ($\sigma<1.0$), automated investigation for moderate uncertainty ($1.0<\sigma<2.0$), and batch review for low confidence ($\sigma>2.0$).

Fit with Neurocomputing

This manuscript aligns perfectly with Neurocomputing's scope and recent publications:

- *Machine Learning Theory:* Rigorous probabilistic framework with theoretical guarantees
- *Real-World Applications:* Addresses critical cybersecurity challenges with practical deployment
- *Computational Efficiency:* Novel sparse approximations enabling large-scale deployment
- *Related Publications:* Similar to recent Neurocomputing papers on uncertainty quantification (e.g., "Deep Gaussian Processes for..." by [Author, Year]) and heterogeneous learning

Reproducibility and Data Availability

Consistent with open science principles:

- The code Publicly available at: https://github.com/rogerpanel/CV/blob/54e68df602d75a2f4ce7143b24204c97b804ea66/hgp-model-v2.ipynb
- *Data:* ICS3D dataset available at Kaggle (DOI: 10.34740/kaggle/dsv/12483891) under CC BY-NC-SA 4.0 license
- *Supplementary Material:* Comprehensive 14-section appendix with all proofs, algorithms, and implementation details.

Manuscript Organization

The 29-page main manuscript follows a logical structure:

- Sections 1-2: Motivation and comprehensive related work (99 references)
- Section 3: Rigorous problem formulation with mathematical foundations
- Section 4: Hierarchical GP methodology with theoretical guarantees
- Section 5: Algorithms and implementation details
- Sections 6-7: Extensive experiments on 21.48M records, outperforming 15 baselines
- Sections 8-9: Discussion and conclusions with future directions

The supplementary appendix (14 sections) provides complete proofs, detailed kernel specifications, extended results, and three case studies demonstrating detection of zero-day MQTT attacks, container escapes, and APT campaigns.

Target Audience

This work will interest multiple Neurocomputing reader communities, such as ML Researchers who are into Novel GP architecture and theoretical contributions; Security Practitioners who are do Practical framework for operational deployment; applied scientists who conduct case studies demonstrating real-world effectiveness; and Method Developers involving in transferable techniques for heterogeneous data.

Declarations

- *Funding:* This work was supported by a grant from the Ministry of Economic Development of the Russian Federation (identifier 000000C313925P3Q0002).
- *Conflicts of Interest:* The authors declare no conflicts of interest whatsoever.
- *Ethics Approval:* Not applicable (publicly available datasets, no human subjects).
- *Data Availability:* Fully available as already detailed above.
- *Author Contributions:* Roger Nick Anaedevha designed the study, developed methods, conducted experiments, and wrote the manuscript. Alexander .Gennadevich Trofimov supervised the work and provided theoretical guidance. Yuri Vladimirovich Borodachev contributed to experimental resources and manuscript revision.
- *Previous Submission:* This manuscript has not been published elsewhere and is not under consideration at another journal.

We believe this manuscript makes significant theoretical and practical contributions to machine learning for cybersecurity, demonstrating how principled uncertainty quantification can address real operational challenges. The rigorous experimental validation on large-scale heterogeneous data, combined with theoretical guarantees and reproducible implementation, aligns well with Neurocomputing's standards for high-quality research.

Thank you for your consideration. We believe the manuscript will be of strong interest to Neurocomputing’s readership, as we look forward to your editorial decision and reviewer feedback.

Sincerely,

Roger Nick Anaedevha (corresponding author)
National Research Nuclear University, MEPhI
Email: rogernickanaedevha@gmail.com | ar006.campus@mephi.ru

Alexander Gennadevich Trofimov
National Research Nuclear University MEPhI

Yuri Vladimirovich Borodachev
Artificial Intelligence Research Center, MEPhI