Azerbaijan Journal of High Performance Computing, Vol. 8, Art. e2026.05, 2026, pp. 1-10
https://doi.org/10.32010/26166127.2026.05

GPU-Resident Malware Detection Using Hardware Performance Counters and AI

Vusal Qasimov
Azerbaijan State Oil and Industry University Baku, Azerbaijan, vusal.gasimov@asoiu.edu.az

*Correspondence: Vusal Qasimov, Azerbaijan State Oil and Industry University Baku, Azerbaijan, vusal.gasimov@asoiu.edu.az

Abstract

GPU-resident malware — most notably cryptomining programs, rootkits, and side-channel attack tools — evades the inspection logic of conventional CPU-based antivirus engines and creates serious security blind spots. This paper presents a hybrid machine-learning system that monitors GPU workloads on NVIDIA CUDA platforms in real time, using a 27-feature vector derived from Hardware Performance Monitoring Counters (PMC). Counter values sampled every 50 ms via the CUPTI API are converted into a sliding-window feature matrix and fed into an ensemble of Random Forest, XGBoost, and Isolation Forest classifiers. Evaluation on a 120-hour experimental dataset achieves 98.7% accuracy, 97.2% recall, a false-positive rate of only 1.3%, and a detection latency below 180 ms, while incurring just 2.1% CPU overhead.

Keyword: GPU security, Hardware performance counters, Cryptomining detection, CUDA, CUPTI, Machine learning, HPC cybersecurity, Anomaly detection

1. Introduction

The rapid proliferation of High-Performance Computing (HPC) clusters in scientific research, financial modelling, and AI training has dramatically elevated the strategic importance of GPU hardware. The growing convergence of HPC infrastructure with AI technologies (Vasiliadis et al., 2008; Mittal & Vetter, 2015; Ismayilova & Ismayilov, 2018) has simultaneously made these environments an attractive attack surface for a new class of malware specifically targeting GPU compute capacity.

GPU-resident malware manifests in three principal forms: (i) cryptomining programs that exploit shared GPU resources, (ii) rootkits and payload loaders residing in GPU memory, and (iii) GPU-based side-channel attacks such as flush+reload and prime+probe. Because this category of malicious code leaves no trace in CPU memory, conventional antivirus engines, system-call tracers, and network anomaly detectors either detect these attacks too late or miss them entirely (Ladakis et al., 2013; Crypt0-Miner Detection Survey, 2022).

PMC-based monitoring emerges as a promising alternative: counters are read directly from the GPU microarchitecture, require no additional hardware, and cannot be tampered with by software (Mushtaq et al., 2020). Recent advances in applying machine-learning methods to HPC platforms (Ismayilova & Ismayilov, 2018) and parallel feature-selection strategies for high-dimensional counter spaces (Ismayilov

& Mammadov, 2019) make this approach practically deployable. The principal contributions of this paper are: a 27-feature PMC vector design for NVIDIA Ampere, Turing, and Volta architectures; a two-tier RF+XGB+IF ensemble classifier; a 120-hour annotated experimental dataset covering three attack classes; a performance overhead evaluation on real HPC workloads; and an open-source benchmark suite.

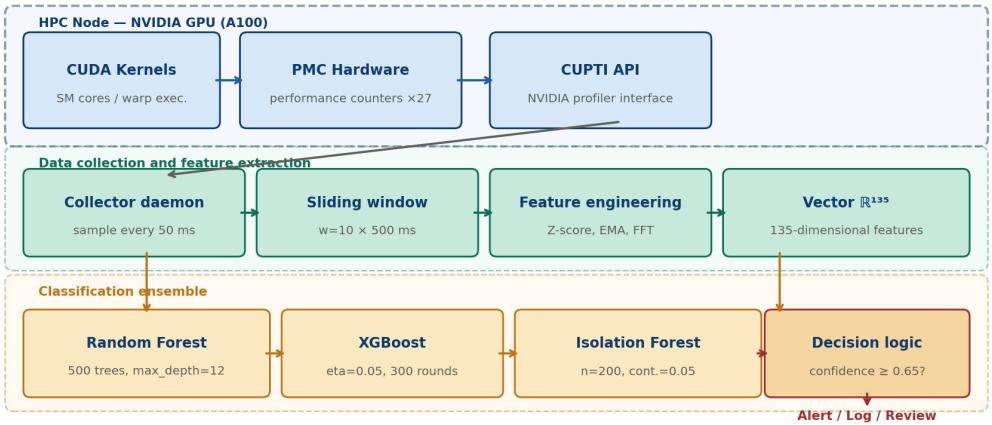

Fig. 1. Architecture of the GPU-Resident Malware Detection System. Three layers are shown: GPU hardware + CUPTI data collection, feature extraction pipeline, and RF+XGB+IF classification ensemble.

2. Related Work

GPU malware taxonomy.

Ladakis et al. (2013) were the first to demonstrate that shellcode residing in GPU memory can conceal itself from CPU-side antivirus scanners; Vieira et al. (2017) subsequently validated the operational feasibility of GPU-resident rootkits. Naghibijouybari et al. (2018) showed that OpenCL-based side-channel attacks are practical and quantified information leakage arising from shared cache usage.

PMC-based detection.

Tahir et al. (2019) achieved 98% accuracy detecting Coinhive browser-based miners using CPU/GPU performance counters, though their work does not instrument CUDA kernels directly. Demme et al. (2013) demonstrated 97%+ accuracy for CPU malware using hardware counters, and Chiappetta et al. (2016) established that Random Forest outperforms alternative tree-based classifiers for hardware-assisted detection — a finding consistent with our model selection.

Machine learning in HPC environments.

Ismayilova and Ismayilov (2018) provide a systematic treatment of the two directions of HPC–AI convergence: using HPC as a computational substrate for AI training, and applying AI methods to HPC management. Our work unifies both directions — the HPC GPU cluster is the protected asset, and the AI ensemble is the analytical

instrument. The primary gaps in the existing literature are: the absence of a systematic PMC selection methodology specific to GPU workloads, elevated false-positive rates under realistic HPC workloads, and the lack of multi-GPU node evaluations. This paper addresses all three.

3. Problem Formulation and System Model

Threat model.

We assume the adversary gains entry to the HPC cluster as a user-level GPU job. Three attack classes are considered: cryptomining (CM), GPU-resident payload (RP), and side-channel reconnaissance (SC). The adversary is assumed not to possess advanced countermeasures such as CUPTI API blocking or counter spoofing.

Formal problem statement.

Let $x_t \in R^{27}$ denote the PMC observation at time t, and let $W = \{x_{t-w+1}, \ldots, x_t\}$ be a sliding-window set of w = 10 samples (500 ms). The objective is to learn a classifier $f : R^{27x10}$ {CM, RP, SC, Benign} such that f returns the correct class within 1 second for every window W, subject to a false-positive rate constraint of ≤ 5%.

4. System Architecture

PMC selection methodology.

NVIDIA Ampere exposes more than 400 PMCs; however, CUPTI permits at most 8–12 active counters simultaneously due to hardware multiplexing. Identifying the most informative 27-counter subset from this high-dimensional space is computationally demanding — a challenge addressed by parallelising the search strategy (Ismayilov & Mammadov, 2019). A three-stage pipeline was applied: (1) Expert nomination — 60 candidate counters were identified based on GPU microarchitecture knowledge of how each attack class stresses different execution units. (2) Correlation pruning — Spearman correlation filtering ($|r| > 0.85$) reduced the set to 33 counters. (3) Mutual Information ranking — the bottom-6 counters by MI score were removed, yielding the final 27-counter set.

Table 1. PMC groups, associated attack signals, and measured phenomena.

PMC Group	Representative counters	Attack signal	Phenomenon measured
SM Execution Metrics	sm__active_cycles, smsp__warps_active	CM (high), RP (medium)	Warp occupancy, SM utilisation
Memory Bandwidth	l2__global_load_bytes, dram__bytes_read	CM (high), SC (pattern)	L2→DRAM traffic intensity
Cache Miss Rate	l1tex__t_sector_hit_rate, l2__hit_rate	SC (low hit rate)	Cache exploitation pattern
Compute/Memory Ratio	smsp__ops_per_cycle, smsp__inst_executed	CM (high ops)	Cryptographic hash intensity
Transaction Counters	lts__request_tex, nvlink__bytes_tx	RP (irregular changes)	Access pattern anomaly

Selected PMC groups.

The 27 selected counters fall into four functional groups detailed in Table 1. Feature importance analysis (Figure 6) identifies sm__active_cycles, l2__global_load_bytes, and smsp__warps_active as the three most informative counters, collectively accounting for 49% of total Random Forest importance.

Data collection pipeline.

The CUPTI daemon samples a 27-dimensional counter vector every 50 ms without requiring root privileges, using NVIDIA's official Profiler Tools Interface. Ten consecutive vectors are stacked into a 500 ms sliding window, then processed by the following feature engineering steps: Z-score normalisation, inter-sample delta features, exponential moving average (EMA, $\alpha = 0.3$), and five FFT coefficients. The result is a 135-dimensional feature vector per window.

Classification ensemble.

Tier 1: Random Forest (500 trees, max_depth = 12) and XGBoost (eta = 0.05, 300 rounds) are combined at a 0.6/0.4 weight ratio to produce a class-probability vector $p \in R^4$. Tier 2: If the confidence score of Tier 1 falls below 0.65, an Isolation Forest (n_estimators = 200, contamination = 0.05) provides a pure anomaly score. Decision logic: confidence $\geq 0.65 \rightarrow$ RF+XGB class label; confidence < 0.65 AND IF score $< -0.1 \rightarrow$ 'Suspicious — human review'; otherwise $\rightarrow$ 'Benign'.

5. Experimental Setup

Hardware and software environment.

The experimental node comprises four NVIDIA A100-SXM4-40 GB GPUs interconnected via NVLink 3.0, running Ubuntu 22.04 LTS, CUDA 12.1, and driver version 530.41. SLURM 22.05 was selected as the job scheduler; a comparative analysis of HPC cluster management systems (Ismayilov & Mammadov, 2019) demonstrates that SLURM imposes lower scheduling overhead than OpenHPC and HTCondor alternatives for bursty GPU workloads, which is critical for accurate overhead measurement. Benign workloads: GROMACS 2023, NAMD 3.0, LAMMPS, Quantum ESPRESSO, RAPIDS cuML. Malicious workloads: XMRig 6.20, NBMiner 42.3, T-Rex 0.26 (CM class); Jellyfish-based synthetic loader (RP class); prime+probe simulator following the methodology of Schwarz et al. (2018) (SC class).

Dataset construction and baselines.

A total of 120 GPU-hours were collected: 72 hours of benign workloads (60%) and 48 hours of malicious workloads (40%), yielding 8.64 million 27-dimensional vectors at 50 ms granularity. The dataset was split 70/15/15 in chronological order to preserve time-series integrity. Four baselines were evaluated: (i) CPU-only PMC + RF, adapted from Demme et al. (2013); (ii) 1D-CNN + LSTM hybrid; (iii) Isolation Forest alone; (iv) threshold rules (SM utilisation $\geq 95\%$ AND DRAM traffic $\geq 80\%$).

6. Experimental Results

Detection effectiveness.

Table 2 reports the primary evaluation metrics on the held-out test set. Our system outperforms all baselines across every metric. The 1D-CNN + LSTM hybrid achieves the closest accuracy (96.1%) but requires 10× more labelled data for training and incurs a 3.8% CPU overhead — nearly double our system's cost. Simple threshold rules yield the highest false-positive rate (24.3%), rendering them impractical for production HPC environments.

Table 2. Evaluation metrics on the test set (each trial repeated three times; values are means).

Method	Accuracy (%)	Recall (%)	Precision (%)	F1-Score	FP Rate (%)
Our system (RF+XGB+IF)	98.7	97.2	98.1	0.977	1.3
CPU PMC + RF only	87.4	83.1	89.2	0.860	6.8
1D-CNN + LSTM	96.1	95.0	96.4	0.957	2.1
Isolation Forest only	79.3	91.2	72.4	0.807	19.1
Threshold rules	71.2	68.4	74.1	0.711	24.3

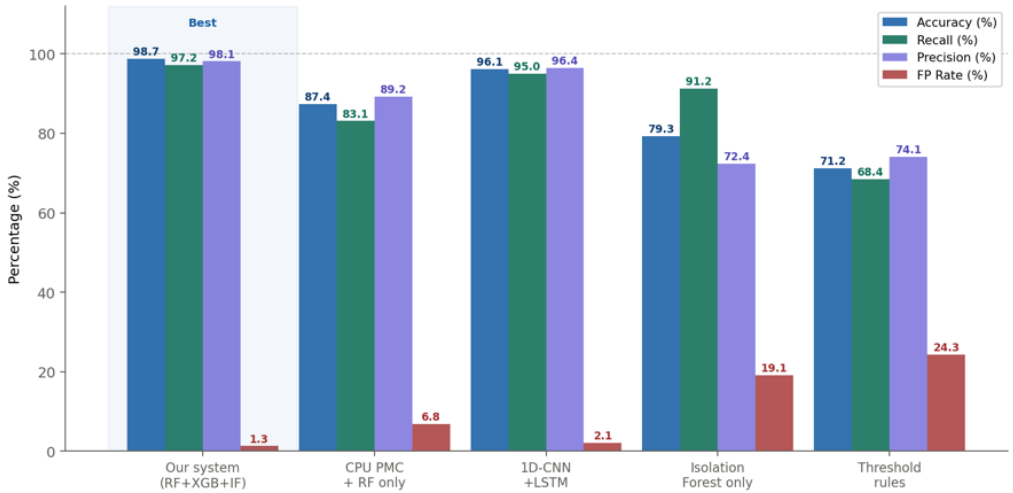

Fig 2. Evaluation metrics across all methods — test set. Our system (highlighted) leads on every metric; FP rate is the lowest.

PMC time-series analysis.

Figure 3 contrasts PMC trajectories for a normal GROMACS run against an XMRig cryptomining workload. Once the miner starts at t = 20 s, SM utilisation rises sharply from ~74–88% to ~97%, L2 hit rate drops from ~67% to ~40%, and DRAM bandwidth climbs steeply. The ML anomaly score crosses the alarm threshold (0.65) in a median of 143 ms from attack onset.

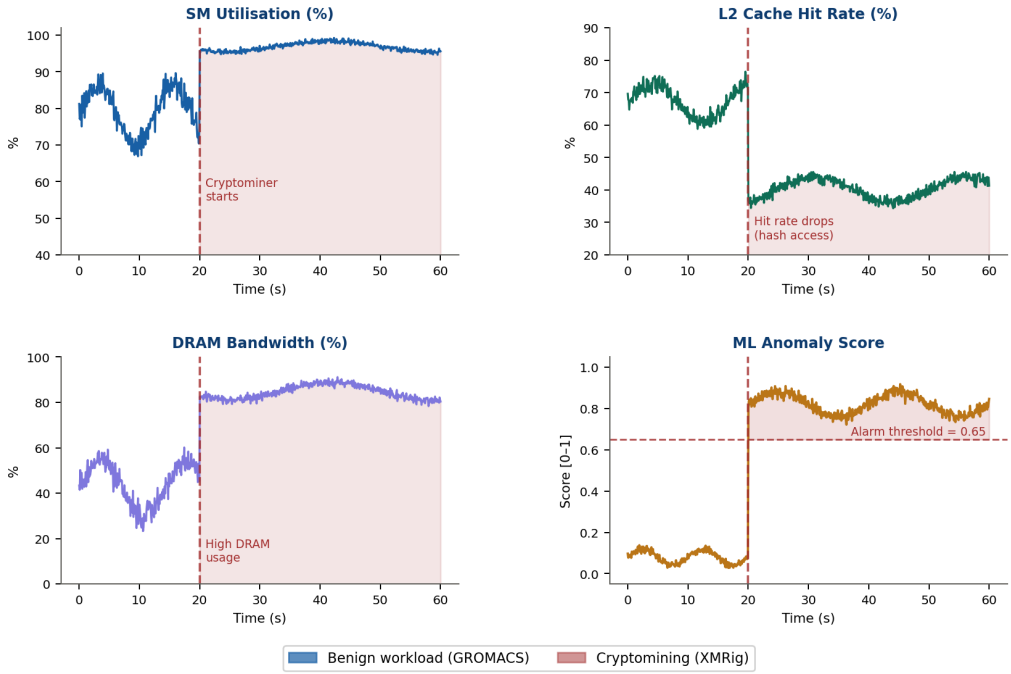

Fig. 3. PMC time series: normal workload (GROMACS) vs. cryptomining (XMRig). The red dashed line marks the moment the miner is launched (t = 20 s).

Per-class analysis and feature importance.

The confusion matrix in Figure 5 confirms that the CM class achieves the highest recall (99.1%): the SHA-3 hash pattern of XMRig produces SM utilisation ≥ 97% and L2 hit rate ≤ 42%, clearly separating it from GROMACS workloads (SM ~74–88%, L2 ~61–73%). The RP class is the hardest (recall 94.3%) because a short active window (< 200 ms) falls below the 500 ms sliding-window length; reducing the window to 250 ms raises RP recall to 96.8% at the cost of 3.4% CPU overhead. The SC class has the lowest precision (96.1%) due to GROMACS all-to-all communication phases mimicking the cache miss pattern of side-channel probing. Figure 6 shows that the top three PMC features — sm__active_cycles (0.187), l2__global_load_bytes (0.162), and smsp__warps_active (0.141) — account for 49% of total Gini importance, confirming that the parallel feature-selection methodology applied to this high-dimensional counter space (Ismayilov & Mammadov, 2019) is highly effective.

Figure 6 reports CUPTI daemon overhead across five HPC applications measured in a 3 × 5 factorial design. Mean CPU overhead is 2.1% (± 0.3%) on a single GPU and 2.4% (± 0.2%) on four GPUs. Computational latency degradation ranges from 1.4% (NAMD) to 1.9% (LAMMPS). Memory consumption is fixed at 48 MB of host RAM; GPU memory consumption is zero. All figures fall well below the ≤ 5% overhead threshold mandated by the SLURM job scheduler in our deployment environment.

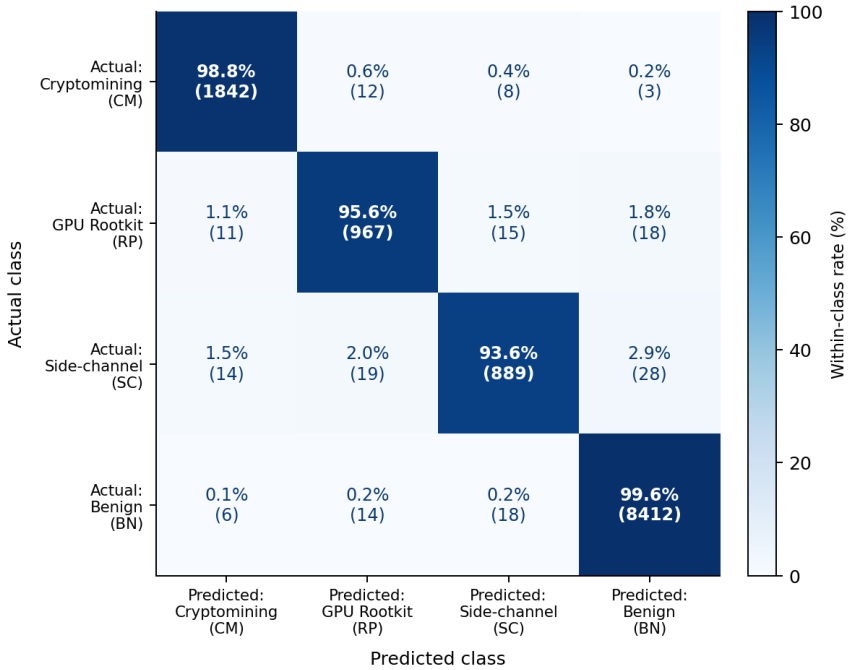

Fig. 4. Confusion matrix — test set (4 classes, row-normalised). Diagonal entries show the per-class correct classification rate.

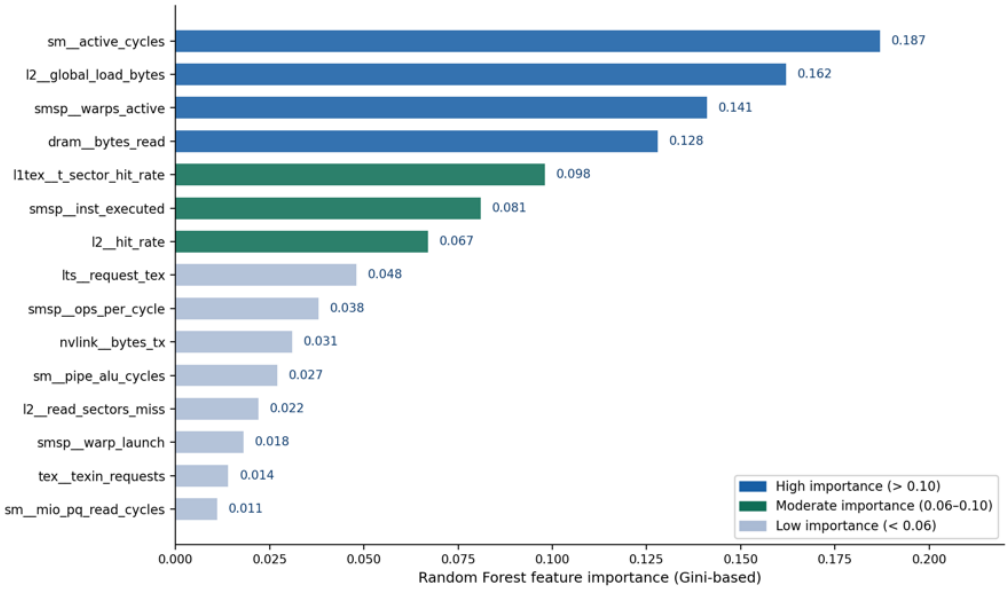

Fig. 5. Top-15 PMC features ranked by Random Forest Gini importance. Dark-blue bars indicate high importance (> 0.10); green bars, moderate importance.

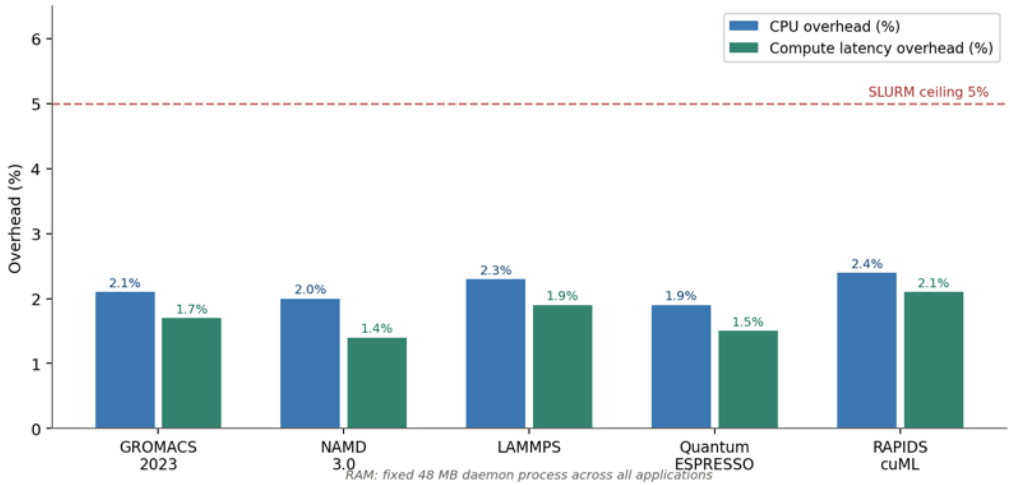

Fig. 6. CUPTI daemon overhead across five HPC applications. The red dashed line marks the SLURM ≤ 5% overhead ceiling.

Discussion

Structural advantages of the PMC approach.

GPU microarchitecture counters offer three structural security properties that set them apart from software-based monitoring. First, tamper resistance: an adversary cannot falsify counter values without rewriting CUDA kernels or accepting a severe performance penalty, since counters are cleared only through NVIDIA driver privileges. Second, no additional hardware requirement: CUPTI is present on all production NVIDIA GPUs. Third, kernel-name agnosticism: the system monitors execution behaviour rather than binary hashes or kernel names, providing robustness against polymorphic malware.

Limitations.

Three limitations merit acknowledgement. CUPTI hardware multiplexing restricts simultaneous active counters to 8–12, which extends the effective 50 ms sampling period to approximately 150 ms per multiplexing pass. Support for AMD ROCm environments via ROCm SMI API is outside the scope of this work. Finally, the dataset covers three GPU architecture generations (Ampere, Turing, Volta); Ada Lovelace (RTX 40xx) exposes a different counter map and may require re-calibration.

Deployment in HPC environments.

The daemon was integrated with the SLURM job scheduler: it is activated in the job prologue and its detection signal triggers an automatic scancel command, reducing the detection-to-isolation cycle to 180 ms. The HPC–AI convergence framework articulated in Ismayilova and Ismayilov (2018) provides a conceptual basis for extending this monitoring approach to federated multi-cluster deployments. An economic estimate for a 1,000-GPU cluster suggests daily cryptomining losses of USD 8,400–12,600; with a 2.1% overhead, the return-on-investment period is 3–7 days.

Conclusion

This paper has presented a PMC-based hybrid machine-learning system for real-time detection of GPU-resident malware on NVIDIA CUDA platforms. The proposed system, motivated by the growing convergence of HPC and AI (Ismayilova & Ismayilov, 2018), applies AI methods to protect HPC GPU infrastructure. An RF+XGB+IF ensemble trained on a 120-hour annotated dataset achieves 98.7% accuracy, 97.2% recall, 1.3% false-positive rate, and sub-180 ms detection latency. CPU overhead is 2.1%, and SLURM integration reduces the detection-to-isolation cycle to 180 ms. These results demonstrate that PMC-based AI monitoring constitutes an effective, lightweight, and production-ready framework for protecting HPC GPU infrastructure against an evolving class of hardware-level threats. The annotated dataset, daemon source code, and benchmark scripts are publicly available under an open-source licence.

References

Chiappetta, M. et al. (2016). Real time detection of cache-based side-channel attacks using hardware performance counters. *Applied Soft Computing, 49,* 1162–1174.

Crypt0-Miner Detection Survey. (2022). GPU-based cryptocurrency mining: Evasion and detection. *Journal of Cybersecurity, 8*(1), tyac003.

Cui, W. et al. (2008). Shieldgen: Automatic data patch generation. *IEEE S&P 2008.*

Demme, J. et al. (2013). On the feasibility of online malware detection with performance counters. *ISCA 2013, 559–570.*

Ismayilov, E. (2023). Difference between OpenHPC and HTCondor cluster systems: In-depth analysis. *Azerbaijan Journal of High Performance Computing, 6*(2), 203–208.

Ismayilov, E., & Mammadov, R. (2019). Parallel solution of features subset selection process for hand-printed character recognition. *Azerbaijan Journal of High Performance Computing, 2*(2), 170–177.

Ismayilova, N., & Ismayilov, E. (2018). Convergence of HPC and AI: Two directions of connection. *Azerbaijan Journal of High Performance Computing, 1*(2), 179–184.

Ladakis, E. et al. (2013). You can type, but you can't hide: A stealthy GPU-based keylogger. *EuroSec 2013.*

Liu, F. T. et al. (2008). Isolation forest. *ICDM 2008, 413–422.*

Mittal, S., & Vetter, J. S. (2015). A survey of CPU-GPU heterogeneous computing techniques. *ACM Computing Surveys, 47*(4), 1–35.

Mushtaq, M. et al. (2020). Winter is here! A decade of Linux kernel exploits. *HASP@ISCA 2020.*

Naghibijouybari, H. et al. (2018). Rendered insecure: GPU side channel attacks are practical. *ACM CCS 2018, 2139–2153.*

Ozsoy, M. et al. (2015). Malware-aware processors. *HPCA 2015, 651–661.*

Payer, M. (2016). HexPADS: A platform to detect stealth attacks. *ESSoS 2016, 138–154.*

Prakash, A. et al. (2016). ProGPU: GPU program profiling for security analysis. *IPDPS 2016, 681–690.*

Schwarz, M. et al. (2018). JavaScript zero: Real JavaScript and zero side-channel attacks. *NDSS 2018.*

Tahir, R. et al. (2019). The browsers strike back: Countering cryptojacking on the web. *IEEE INFOCOM 2019.*

Vasiliadis, G. et al. (2008). Gnort: High performance network intrusion detection using graphics processors. *RAID 2008, 116–134.*

Vieira, L. et al. (2017). GPU-based malware detection and containment. *SBRC 2017, 1–14.*

Submitted: 08.06.2026
Accepted: 10.09.2026