Early Detection of Ransomware via Multi-Source Fusion with Hardware Performance Counters

Yonghui Xi, Siyu Zhang, Jun Song, and Fan Yang(✉)

China University of Geosciences, Wuhan, China
fanyang@cug.edu.cn

Abstract. In recent years, ransomware detection based on hardware performance counters (HPCs) has shown promise. However, existing HPC-based approaches are constrained by limited feature modeling and insufficient behavioral context, leading to high false positive rates and detection delays from data accumulation. To address these issues, we propose an early detection approach that combines HPCs, disk input/output (I/O) statistics, and ransom note analysis. The approach introduces an attention-enhanced temporal convolutional network to extract multi-scale features from HPC data, thus improving the extraction of temporal characteristics in HPC sequences. Furthermore, disk I/O patterns are analyzed through machine learning to provide additional behavioral context beyond HPCs, significantly reducing false positives. We also design a Minifilter-based ransom note detector to identify ransom notes dropped by ransomware, further reducing detection latency. Extensive experiments demonstrate that our approach achieves a Matthews Correlation Coefficient (MCC) of 95.95%, a false positive rate of only 0.48% on unknown ransomware, and reduces average detection latency by 0.8 seconds across 14 ransomware families. It outperforms representative approaches and exhibits strong potential for detecting unknown ransomware.

Keywords: Ransomware Detection, Hardware Performance Counters, Temporal Convolutional Network, Ransom Notes.

1 Introduction

Ransomware is a constantly evolving threat that can cause rapid and extensive damage to individuals and organizations. The widespread availability of ransomware development toolkits has enabled attackers to generate a multitude of novel and evasive variants with relative ease. Furthermore, the advent of Ransomware-as-a-Service (RaaS) has significantly lowered the technical entry barrier, contributing to a dramatic rise in ransomware incidents [1]. A recent study reported 7,515 publicly disclosed ransomware victim organizations in 2025, representing a 58% increase over the previous year [2]. These alarming figures highlight the urgent need to design effective ransomware detection techniques capable of keeping pace with the evolving threat landscape.

Ransomware detection is an ongoing arms race, where researchers design novel methods as adversaries continually refine their tactics to evade them. Signature-based antivirus software (AVS) remains one of the most widely used tools in commercial

antivirus products [3]. It identifies ransomware by matching an executable's signature against previously stored ransomware signatures. However, signature-based AVS fails to detect unknown ransomware, that is, previously unseen ransomware, because the corresponding signatures are absent from the database. In fact, it is not effective even against known ransomware with polymorphic or metamorphic characteristics [4]. Such ransomware either employs a mutation engine or rewrites its code via advanced obfuscation techniques. Although behavior-based AVS is promising for detecting unknown and morphing ransomware, it incurs substantial computational overhead [5]. As a result, monitoring system-wide activity in real-time becomes challenging, particularly in environments with intensive process activity.

Recent studies have investigated hardware-assisted ransomware detection by leveraging hardware as a root of trust. The rationale is that, although antivirus software can be evaded through code-level variations, hardware-based detectors are harder to circumvent because of the inherent tamper-resistance of hardware [6, 7]. In this context, hardware performance counters (HPCs) have emerged as a promising detection signal. HPCs are specialized registers in the Performance Monitoring Unit (PMU) of modern processors that record hardware events generated during program execution [8]. Compared with software-based profilers, HPC-based monitoring introduces lower overhead in collecting runtime information [6]. However, HPC-based ransomware detection is still at an early stage. Existing studies have applied machine learning to HPC data for ransomware detection [9-15], but these methods face two fundamental limitations: (1) they usually rely on single-scale features without suppressing irrelevant ones, which weakens feature extraction; and (2) their sole dependence on HPCs provides insufficient behavioral context, causing high false positives and detection delays due to the need for runtime data accumulation.

In this paper, we propose a novel HPC-based multi-source collaborative approach for ransomware detection. The approach integrates the monitoring of HPCs, disk I/O, and ransom notes to enable early detection of ransomware attacks. It explores the potential of multi-scale temporal convolutional networks (TCNs) to extract crucial temporal features from HPCs, disk I/O analysis to reduce false positives, and ransom note monitoring to trigger early alerts. The main contributions are as follows:

1. We propose an attention-integrated multi-scale TCN (SE-TCN) for HPC-based ransomware detection, which enhances temporal modeling capability and improves detection performance, achieving MCC gains of 3.80%–23.41%.
2. We introduce disk I/O data as an additional information source to compensate for the limited behavioral context of HPCs. This provides richer behavioral evidence for ransomware detection and effectively reduces false positives.
3. We design a Minifilter-based ransom note detection mechanism with real-time file system monitoring, enabling earlier ransomware detection and reducing the average detection latency by 0.8 seconds across 14 representative families.

2 Related Work

The arms race between ransomware attacks and ransomware detection has persisted for more than two decades. In the early stages, detection efforts focused on static analysis [16]. The fundamental idea of static analysis is to extract rule-based features using expert domain knowledge, and then detect ransomware through signature matching or machine learning algorithms. Unfortunately, this conventional approach can be evaded through obfuscation [17] and fileless techniques [18]. Dynamic detection techniques try to counter such evasion techniques. Instead of relying on static features or signatures, these methods analyze the runtime behavior of ransomware to distinguish it from benign software. However, dynamic detection techniques typically focus on process-level or file-level behaviors of ransomware, which often incur high system overhead [12]. Consequently, AVS struggles to efficiently detect ransomware exhibiting strong obfuscation or other evasion capabilities.

Researchers have recently shifted their focus toward HPC-based detection approaches, which demonstrate stronger resistance to ransomware attacks and lower system overhead compared to traditional software-based approaches [6, 7]. Existing HPC-based detection approaches can be broadly categorized into unsupervised anomaly detection and supervised classification. Representative unsupervised methods, such as RATAFIA [9] and HARD-Lite [10], leverage Long Short-Term Memory (LSTM) networks to learn the temporal patterns of benign HPC data and detect ransomware by identifying significant deviations from normal behavior. However, these approaches struggle to identify novel or multi-process ransomware variants, and lowering detection thresholds to improve sensitivity inevitably increases the false positive rate. Although RATAFIA introduces a two-stage verification mechanism, both stages rely solely on HPC data, yielding only limited improvement.

Supervised methods [11-13] aim to learn ransomware patterns from HPC data. DeepWare [12] employs CNNs for feature extraction and achieves high detection accuracy across multiple ransomware families, but its focus on single-scale temporal features and limited ability to suppress irrelevant features constrain performance. HiPeR [13] explores HPC features related to non-encryption behaviors to improve early detection, yet these features remain susceptible to interference from unrelated system activity. Overall, existing methods suffer from two major limitations: they rely on single-scale feature extraction with limited suppression of irrelevant features, and they depend solely on accumulated low-level HPC data without behavioral context, leading to false positives and detection delays. To address these issues, this work designs an attention-enhanced multi-scale TCN model for HPC feature extraction. Unlike RATAFIA, which uses dual-stage verification based only on HPCs, the proposed approach further incorporates disk I/O features to reduce false positives and introduces real-time ransom note detection as an additional indicator to enable earlier detection.

3 Proposed Approach

The proposed approach leverages the complementary strengths of HPC monitoring, disk I/O statistics, and real-time ransom note detection to enable efficient multi-perspective characterization of ransomware behavior (see Figure 1). By jointly analyzing data from both hardware events and disk activity, the system captures reliable behavioral features indicative of ransomware attacks. Two specialized machine learning classifiers are trained on these heterogeneous data streams, and their outputs are fused at the decision level using a consensus-based voting scheme, effectively reducing false positives. Furthermore, ransom note detection is performed through continuous I/O monitoring and targeted content analysis of relevant files, further accelerating response to ransomware. The following sections detail each component.

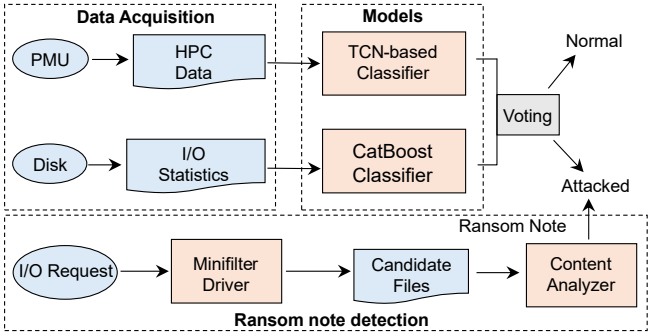

Fig. 1. The proposed ransomware detection approach consists of three components: (1) multi-source data acquisition, (2) classification models, and (3) ransom note detection.

3.1 Multi-Source Data Acquisition

Accurately characterizing ransomware behavior requires multi-perspective data acquisition, as a single data source cannot fully capture its complexity. Existing HPC-based methods [9-13] can detect low-level microarchitectural anomalies, but often yield high false positive rates when distinguishing ransomware from benign applications with intensive resource usage because they lack high-level behavioral indicators. Disk I/O metrics reflect ransomware-related file-access intensity and patterns, but on their own cannot reliably distinguish malicious encryption from legitimate disk-intensive operations such as backups. To address these limitations, this work collects both HPC data and disk I/O statistics, combining hardware-level and block-device-level signals to capture ransomware-related anomalies across layers and improve detection accuracy.

To balance detection accuracy and system overhead, different sampling intervals are adopted for the two data sources. HPC data are sampled every 10 milliseconds to enable fine-grained real-time tracking of behavioral changes. This interval is consistent with prior work [9, 12] and close to the threshold at which monitoring overhead increases sharply. Disk I/O statistics are aggregated every 200 milliseconds. Although a previous study [14] used intervals as short as 20 milliseconds, overly frequent disk sampling may increase overhead and affect data reliability, thereby reducing feature quality. The

sampled data are then organized into time-windowed sequences to model temporal dependencies.

Due to the limited number of HPC events that can be monitored simultaneously, this work prioritizes events closely related to file traversal and encryption, including instructions, branches, branch misses, cache references, and cache misses. For disk I/O, this work selects metrics that best reflect file-access intensity during encryption, including the number of read requests (rd_req), bytes read (rd_bytes), write requests (wr_req), bytes written (wr_bytes), total read time (rd_total_time), and total write time (wr_total_time), while excluding features such as flush_total_time that mainly reflect normal synchronization activity. This design balances hardware constraints with accurate behavioral representation, supporting efficient multi-source detection.

3.2 Classification Models

The proposed approach employs two classifiers: a classifier based on SE-TCN for HPC data and CatBoost for disk I/O statistics. The former captures temporal patterns in HPC sequences, while the latter operates on relatively simple, interpretable I/O features. Their predictions are integrated using a strict consensus voting scheme that flags an anomaly only when both classifiers concur, thereby reducing false positives.

Classifier for HPC Data. HPC-based ransomware detection is challenging because HPC sequences exhibit complex temporal dependencies. Effective detection requires multi-scale temporal modeling, as ransomware behavior exhibits short-term fluctuations and longer-term periodic patterns across different time scales. Furthermore, feature channels contribute unequally to classification and should be adaptively recalibrated. To address these issues, this work proposes an SE-TCN-based classifier, as illustrated in Figure 2.

SE-TCN uses dilated convolutions to extract multi-scale temporal features from HPC data. By inserting holes into the convolution kernel, dilated convolutions expand the temporal receptive field without significantly increasing the number of parameters or reducing feature-map resolution. Stacking dilated convolutions enables progressive multi-scale temporal modeling. Under unit stride and length-preserving padding, the receptive field after layer i is

$$R_i = 1 + \sum_{j=1}^{i} (k-1)\, d_j, \tag{1}$$

where k is the kernel size and d_j is the dilation rate of layer j.

To model channel importance, SE-TCN introduces the Squeeze-and-Excitation (SE) attention mechanism [19]. Although dilated convolutions capture multi-scale temporal features, they assign equal importance to all channels, including those less relevant to the task. SE addresses this limitation by adaptively recalibrating channel responses. Given an input feature map $F \in \mathbb{R}^{C\times T}$, where C is the number of channels and T is the temporal length, channel-wise attention weights are computed as

$$w = \sigma\big(W_2\delta(W_1 z)\big), \quad z_c = \frac{1}{T}\sum_{t=1}^{T} F_{c,t}\,, \tag{2}$$

where $z \in \mathbb{R}^C$ is obtained by global average pooling along the temporal dimension, W_1 and W_2 are weight matrices, δ denotes ReLU, and σ denotes Sigmoid. The feature map is recalibrated by

$$F'_{c,t} = w_c \cdot F_{c,t}. \tag{3}$$

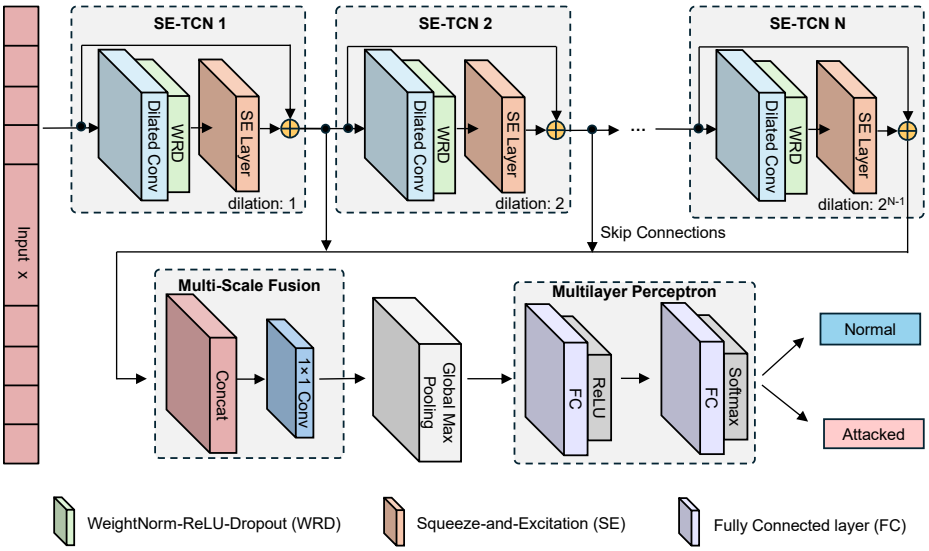

Fig. 2. SE-TCN-based classifier.

To enhance cross-scale fusion, skip connections pass outputs from layers with different dilation rates to a fusion layer, where they are concatenated and fused by a 1×1 convolution. This design promotes feature reuse and improves gradient flow. Max pooling reduces the parameter burden of the fully connected layers, and a softmax layer produces the final binary output. Bayesian optimization is adopted during training.

Classifier for Disk Activity. CatBoost, as a tree-based model, is a natural and effective choice for detecting ransomware from disk I/O statistics. Disk I/O features are sampled every 200 ms and organized as conventional tabular data. Unlike high-frequency HPC sequences, these features do not exhibit strong temporal dependencies. In this case, deep learning models tailored for sequential or high-dimensional data offer limited benefits, whereas tree-based algorithms are better suited [20]. With its ordered boosting mechanism and symmetric tree structure, CatBoost offers strong generalization and robustness to overfitting. Therefore, CatBoost is used to distinguish ransomware-related disk activity from benign operations, ensuring stable classification performance.

To mitigate the risk of increased false negatives induced by enforcing strict decision-level consensus with disk I/O statistics, model optimization places greater emphasis on recall for the ransomware class. This is achieved by using weighted binary cross-entropy loss, where positive samples corresponding to ransomware are assigned larger weights. The loss function is defined as follows:

$$\mathcal{L}_{\mathrm{WCE}} = -\big(\omega_1 \cdot y \cdot \log(\hat{p}) + \omega_0 \cdot (1-y) \cdot \log(1-\hat{p})\big), \tag{4}$$

where y denotes the true label, $\hat{p}$ is the predicted probability, and ω_1, ω_0 represent the weights for the positive and negative classes, respectively. Assigning a higher weight to the positive class effectively increases the model's sensitivity to ransomware, thereby compensating for potential false negatives arising from the strict voting scheme.

3.3 Ransom Note Detection

Ransom note detection provides an effective way to accelerate ransomware detection. Classifiers based on HPC and disk I/O statistics require runtime data accumulation and therefore incur detection latency. Notably, some ransomware families drop ransom notes early during file traversal or initial encryption. Therefore, ransom notes can serve as an early indicator of malicious activity [21]. Specifically, I/O requests trigger asynchronous content inspection through a detector built on Minifilter, which is a kernel-level file system filter driver framework. If ransom note content is identified, the attack is flagged immediately without waiting for HPC or disk I/O data to accumulate.

The detector uses empirical characteristics derived from public ransom note datasets [22, 23] to filter candidate files. Since ransom notes are usually stored in accessible formats $\mathcal{E} = \{.\mathrm{txt}, .\mathrm{html}, .\mathrm{htm}, .\mathrm{hta}\}$ and are typically small [24], each written file f is represented by its extension $\mathrm{ext}(f)$ and size $s(f)$, and is included in the candidate set $\mathcal{F}_c$ only if:

$$f \in \mathcal{F}_c \Leftrightarrow (\mathrm{ext}(f) \in \mathcal{E}) \wedge (s(f) < S_{\max}), \tag{5}$$

where $S_{\max}$ is the maximum file size threshold (75 KB, 99th percentile of ransom notes). This pre-filtering reduces unnecessary content analysis and overhead, while asynchronous processing preserves detection timeliness and system stability.

To infer ransom intent, the detector performs keyword-based content analysis using three keyword categories extracted from public datasets: target identifiers $\mathcal{T} = \{\mathrm{file}, \mathrm{data}\}$, threat descriptions $\mathcal{D} = \{\mathrm{crypt}, \mathrm{lock}, \mathrm{restore}, \mathrm{recover}\}$, and contact information $\mathcal{C} = \{\mathrm{http}, \mathrm{email}, \mathrm{onion}, \mathrm{bitcoin}\}$. For a candidate file with text content X, the detection function is defined as:

$$K(X) = \mathrm{True} \Leftrightarrow \forall \theta \in \{\mathcal{T}, \mathcal{D}, \mathcal{C}\}, \theta \cap X \neq \emptyset. \tag{6}$$

That is, at least one keyword from each category must appear in X. To further improve accuracy, keywords from $\mathcal{T}$ and $\mathcal{D}$ must also co-occur within a character distance, i.e., $d(\mathcal{T}, \mathcal{D}) < \delta$, where δ = 230 corresponds to the 99th percentile in public datasets. This contextual constraint suppresses spurious matches and flags only semantically coherent ransom notes.

4 Experiments

4.1 Experimental Setup

Environment. Experiments were conducted on a desktop equipped with an Intel Core i3-9100T CPU and 16 GB RAM, using Debian 12 as the host OS and a Windows 10 VM as the test environment. To simulate realistic user scenarios, 61.9 GB of user files (68,752 files) were deployed in the VM.

Baselines and Benchmark. We compared the proposed approach with four representative HPC-based ransomware detectors: RATAFIA [9], DeepWare [12], HiPeR [13], and EGB [11]. Pan et al. [15] was not reproduced due to hardware limitations. To assess the contribution of disk I/O data, we also evaluated standard machine learning classifiers on I/O features alone and augmented the baselines with disk I/O features (e.g., DeepWare+, HiPeR+). The ransomware benchmark consists of samples collected from VirusShare [25], MalwareBazaar [26], and HybridAnalysis [27], covering 28 families mainly active between 2021 and 2025, as listed in Table 1. The benign benchmark includes common applications.

Table 1. Ransomware families and the number of samples.

Family	Number of Samples	Family	Number of Samples
AtomSilo	2	Intercobros	1
AvosLocker	14	LockBit	8
BianLian	2	Lorenz	1
BlackBasta	12	Makop	3
BlackMatter	11	Mallox	8
Blackout	4	Mespinoza	15
Cl0p	2	MountLocker	1
Conti	5	Phobos	3
Cuba	4	Play	5
DemonWare	1	Ryuk	6
Dharma	17	Sodinokibi	21
GlobeImposter	2	Surtr	8
HelloXD	1	TeslaCrypt	1
HydraCrypt	1	Zeppelin	1

Data Acquisition and Evaluation. HPC data were sampled every 10 ms and disk I/O statistics every 200 ms. The collected traces were segmented into sliding windows and labeled as ransomware or benign, yielding 86,400 balanced samples. To evaluate generalization, 9 ransomware families (Phobos, GlobeImposter, Cl0p, BianLian, Zeppelin, Blackout, Conti, HydraCrypt, and BlackBasta) were reserved for unknown-family

testing and the remaining families were used for known-family training and testing. We report Acc, Prec, Rec, F1, FNR, FPR, and MCC, with MCC as the primary metric because it accounts for all four confusion-matrix outcomes.

4.2 Detection Performance

We evaluated the proposed model against representative baselines, as shown in Table 2. The proposed model achieves a precision above 99% while maintaining an FPR of only 0.66%, corresponding to relative FPR reductions of 57%, 69%, 60%, and 69% compared with RATAFIA, HiPeR, EGB, and DeepWare, respectively. This improvement mainly comes from constraining the decision with file-access behavioral context, which effectively suppresses false positives. The I/O-only results demonstrate the discriminative value of disk access patterns, while the enhanced variants further show that incorporating I/O features consistently reduces false positives across baselines. Although RATAFIA+ attains a lower FPR, it does so at the cost of a much higher FNR of 30%, because as an unsupervised method it models only normal HPC patterns and cannot accurately distinguish benign from ransomware behaviors. In terms of F1 and MCC, EGB+ and HiPeR+ are close to the proposed model, and DeepWare+ ranks second. The proposed model achieves the best F1 and MCC, benefiting from both the integration of disk I/O features and the enhanced TCN architecture for multi-scale temporal feature extraction.

Table 2. Detection performance (%) on known ransomware families, with the best metrics highlighted in bold.

Source	Model	Acc	Prec	Rec	F1	FNR	FPR	MCC
HPC	DeepWare	98.25	97.87	**98.64**	98.25	**1.36**	2.15	96.50
	HiPeR	96.73	97.77	95.64	96.69	4.36	2.18	93.49
	RATAFIA	84.14	97.86	69.84	81.51	30.16	1.53	71.30
	EGB	97.43	98.29	96.54	97.41	3.46	1.68	94.88
I/O	CatBoost	97.92	98.59	97.24	97.91	2.76	1.39	95.86
	KNN	96.95	97.92	95.93	96.92	4.07	2.04	93.91
	RF	97.21	98.06	96.32	97.18	3.68	1.91	94.43
	Logistic	88.57	82.53	97.87	89.55	2.13	20.72	78.52
	SVM	97.12	98.73	95.47	97.07	4.53	1.23	94.29
HPC+I/O	DeepWare+	98.59	99.15	98.03	98.59	1.97	0.85	97.19
	HiPeR+	96.83	98.99	94.64	96.76	5.36	0.97	93.76
	RATAFIA+	84.51	**99.44**	69.41	81.75	30.59	**0.39**	72.40
	EGB+	97.00	99.01	94.96	96.94	5.04	0.95	94.09
	Ours	**98.98**	99.33	98.62	**98.98**	1.38	0.66	**97.96**

4.3 Generalization Performance

We evaluated model generalization on unknown ransomware families. As shown in Table 3, HPC-only methods (e.g., DeepWare, HiPeR, and EGB) achieved limited recall and MCC, indicating weak generalization. This is because HPC features are highly sensitive to execution-pattern variations: different families exhibit different instruction sequences and hardware event distributions, making such models prone to overfitting to seen families. In contrast, I/O-only methods (e.g., CatBoost and RF) showed relatively stable performance, since disk access patterns reflect family-independent ransomware behaviors such as intensive encryption-related writes. However, they still have limitations. For example, logistic regression reached the highest recall (98.24%) but also produced an extremely high FPR (19.00%), leading to a much lower MCC, suggesting that a linear decision boundary cannot balance sensitivity and specificity well in the current I/O feature space. Fusion models (HPC+I/O) showed the strongest generalization performance. In particular, the proposed approach achieved 96.38% recall, 0.48% FPR, and the highest MCC of 95.95%. This benefit comes from feature complementarity: the HPC branch captures multi-scale temporal patterns while attention suppresses irrelevant channels, and the I/O branch provides more stable high-level behavioral cues. Their combination enables robust modeling of family-independent ransomware behaviors.

Table 3. Detection performance (%) on unknown ransomware families, with the best metrics highlighted in bold.

Source	Model	Acc	Prec	Rec	F1	FNR	FPR	MCC
HPC	DeepWare	95.86	97.27	94.10	95.65	5.90	2.65	91.51
	HiPeR	90.45	93.90	86.53	90.06	13.47	5.62	81.16
	RATAFIA	83.28	97.44	68.36	80.35	31.64	1.79	69.74
	EGB	92.80	96.49	88.83	92.50	11.17	3.23	85.86
I/O	CatBoost	97.06	98.62	95.45	97.01	4.55	1.33	94.17
	KNN	94.11	97.77	90.29	93.88	9.71	2.06	88.49
	RF	96.24	97.73	94.67	96.18	5.33	2.20	92.52
	Logistic	89.62	83.80	**98.24**	90.44	**1.76**	19.00	80.45
	SVM	93.70	98.24	88.99	93.39	11.01	1.60	87.78
HPC+I/O	DeepWare+	96.46	99.42	93.47	96.35	6.53	0.55	93.10
	HiPeR+	92.46	98.66	86.08	91.95	13.92	1.17	85.62
	RATAFIA+	83.67	98.84	68.13	80.66	31.86	0.80	70.84
	EGB+	93.11	98.55	87.51	92.71	12.49	1.28	86.77
	Ours	**98.39**	**99.51**	96.38	**97.91**	3.62	**0.48**	**95.95**

4.4 Detection Latency

To evaluate detection latency, we selected one representative sample from each of 28 ransomware families. Table 4 compares the latency with and without ransom note detection. The results show that 12 families were detected within 1 second, including 6 with near-zero latency, mainly because these variants release ransom notes immediately without a long pre-encryption stage. In contrast, some families, such as AtomSilo, required more than 15 seconds due to evasive actions before file traversal and encryption. Among the 28 families, 14 were detected faster when ransom note detection was enabled, reducing the average latency by 0.8 seconds, since this mechanism does not require sufficient accumulation of HPC and disk data. This benefit is not universal, as some families release ransom notes only in later stages of encryption. Overall, the proposed approach enables early ransomware detection, and ransom note detection can further reduce latency.

Table 4. Detection latency (seconds) across ransomware families without/with ransom note detection.

Family	Detection Latency (s)		Family	Detection Latency (s)	
	Without	With		Without	With
AtomSilo	23.4	22.7	Intercobros	0.8	0.8
AvosLocker	1.2	**< 0.1**	LockBit	3.4	3.4
BianLian	0.6	**< 0.1**	Lorenz	1.0	1.0
BlackMatter	1.0	0.7	Makop	16.4	16.4
BlackBasta	0.8	**< 0.1**	Mallox	0.6	**< 0.1**
Blackout	7.0	7.0	Mespinoza	0.6	0.6
Cl0p	12.4	12.2	MountLocker	2.6	2.6
Conti	4.8	4.8	Phobos	1.0	1.0
Cuba	15.8	15.6	Play	13.0	12.4
DemonWare	3.4	3.4	Ryuk	3.2	1.6
Dharma	0.8	0.8	Sodinokibi	3.0	**< 0.1**
GlobeImposter	2.0	2.0	Surtr	9.8	9.8
HelloXD	14.8	14.5	TeslaCrypt	3.8	3.8
HydraCrypt	0.8	**< 0.1**	Zeppelin	18.8	18.3

4.5 Ablation Study

To evaluate the effectiveness of the proposed model design, we conducted ablation experiments on both known and unknown ransomware families. As illustrated in Figure 3, we started with a CNN baseline and progressively incorporated dilated convolutions (+Dilated), attention mechanisms (+Attention), and skip connections (+Skip Conn.). These architectural enhancements consistently improved detection performance, with the MCC increasing from about 95% to above 97% on known families and from about

91% to above 95% on unknown families. These results demonstrate that the proposed model significantly enhances detection capability while maintaining strong generalization to previously unseen ransomware families. Finally, when disk I/O features were integrated via CatBoost (+CatBoost) under the consensus-based voting scheme, the FPR on unknown families decreased markedly. This suggests that incorporating file access patterns as behavioral context helps focus detection on scenarios involving intensive file activity, thereby reducing false alarms.

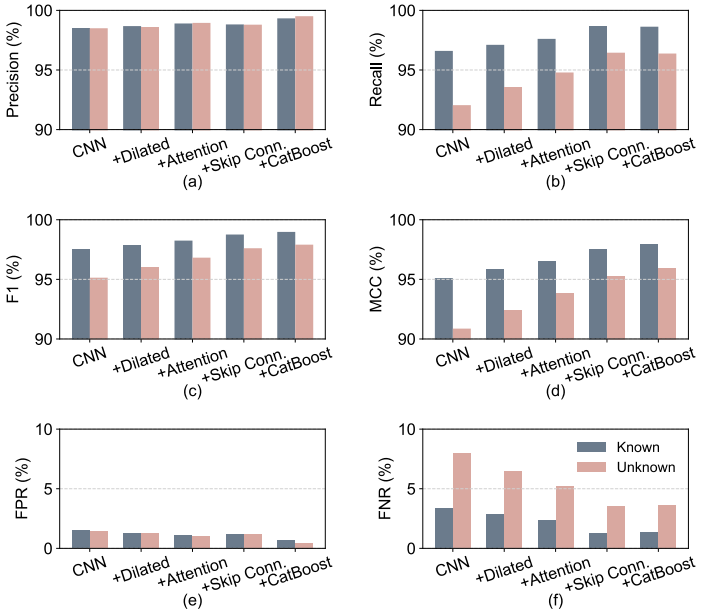

Fig. 3. Ablation study results on known and unknown ransomware, with model components added incrementally.

4.6 Impact of Window Size

Figure 4 shows the effect of different window sizes on performance. Overall, the performance improves as the window size increases. When the window size is set to 200 ms, the model underperforms, as disk activity features within such a short time window limit the overall detection performance. Specifically, ransomware may exhibit intermittent encryption, causing some 200 ms windows to contain little or no disk activity. Meanwhile, benign applications can generate short bursts of intensive I/O during initialization or loading, making benign and malicious behaviors difficult to distinguish within such a narrow window. Expanding the window from 200 ms to 600 ms increases the MCC by about 5%, and further extending it to 1000 ms provides only a marginal gain of less than 1%. However, larger windows also result in longer data accumulation time and, consequently, increased detection latency. Therefore, a 600 ms window is considered optimal for ransomware detection.

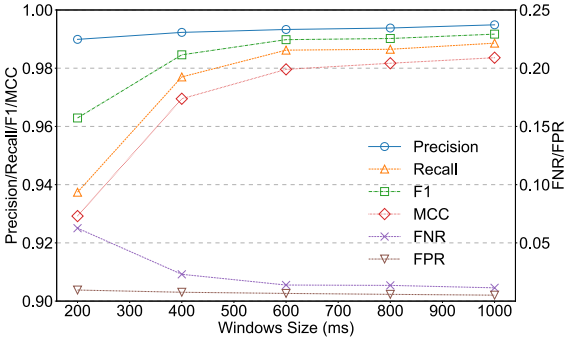

Fig. 4. Impact of window size on model performance.

4.7 Feature Importance

The proposed approach uses five HPC events and six disk I/O metrics for ransomware detection. To assess their contributions, we measured the importance of HPC events using permutation importance [28] based on validation error rate, and obtained disk I/O importance from CatBoost. As shown in Table 5, the HPC model mainly relies on Cache-references and Branch-misses, whereas Instructions and Branches have importance scores close to 1, indicating limited contribution. Thus, when HPC resources are limited, these two events may be assigned lower priority. For disk I/O metrics, CatBoost mainly depends on rd_total_time, wr_total_time, and rd_bytes. Among them, read-related features are more important than write-related ones. This may be because ransomware typically reads files to identify targets before encryption, making read patterns relatively consistent across families. In contrast, write patterns vary more across families because ransomware families differ in overwrite policy, copy-based encryption strategy, and chunk size.

Table 5. Importance of HPC events and disk I/O metrics.

HPC events		Disk I/O metrics	
Event	Importance	Metric	Importance
Cache-references	7.16	rd_total_time	23.66
Branch-misses	7.09	wr_total_time	23.80
Cache-misses	3.69	rd_bytes	19.88
Instructions	1.78	wr_bytes	13.80
Branches	1.72	rd_req	11.95
		wr_req	5.91

5 Conclusion

HPC-based approaches have shown significant potential in ransomware detection but still face challenges, including high false positive rates and detection delays. To address these challenges, we propose an early ransomware detection approach based on multi-source fusion, integrating HPC data, disk I/O activity, and ransom note monitoring. An attention-enhanced TCN extracts multi-scale HPC features, while a CatBoost classifier analyzes disk I/O behavior to restrict positive decisions to scenarios involving intensive file access, effectively reducing false positives. Furthermore, ransom note detection effectively reduces detection latency. Experimental results demonstrate that our approach outperforms representative baselines and reliably detects unknown ransomware families. In future work, we will explore ransom note detection methods that are more robust to paraphrasing and spelling perturbations, and develop a more flexible fusion strategy.

References

1. Fisher, T., Pieri, Z., Howell, C.J., O'Malley, R., Tremblay, L., Dawoud, M.: Vendor communication themes in darknet ransomware-as-a-service (raas) advertisements. Computers in Human Behavior 165, 108571 (2025)
2. GuidePoint Security: GRIT 2026 Ransomware & Cyber Threat Report. https://www.guidepointsecurity.com/resources/grit-2026-ransomware-and-cyber-threat-report, last accessed 2026/01/22
3. Tavares-Silva, S.H.M., Lopes-Lima, S.M., Paranhos-Pinheiro, R., Santiago-Abreu, L.M., Toscano-Lima, R.D., Fernandes, S.M.M.: Antivirus solution to IoT malware detection with authorial next-generation sandbox. The Journal of Supercomputing 81, 151 (2025)
4. Zhang, H., Zhao, L., Yu, A., Cai, L., Meng, D.: Ranker: Early ransomware detection through kernel-level behavioral analysis. IEEE Transactions on Information Forensics and Security 19, 6113–6127 (2024)
5. Kumar, H., Chakraborty, B., Sharma, S., Mukhopadhyay, S.: Xmd: An expansive hardware-telemetry-based mobile malware detector for endpoint detection. IEEE Transactions on Information Forensics and Security 18, 5906–5919 (2023)
6. Pan, Z., Sheldon, J., Mishra, P.: Hardware-assisted malware detection and localization using explainable machine learning. IEEE Transactions on Computers 71(12), 3308–3321 (2022)
7. Cheng, B., Leal, E.A., Zhang, H., Ming, J.: On the feasibility of malware unpacking via hardware-assisted loop profiling. In: 32nd USENIX Security Symposium (USENIX Security 23), pp. 7481–7498. USENIX Association, Anaheim, CA (2023)
8. Sayadi, H., He, Z., Makrani, H.M., Homayoun, H.: Intelligent malware detection based on hardware performance counters: A comprehensive survey. In: 2024 25th International Symposium on Quality Electronic Design (ISQED), pp. 1–10. IEEE, San Francisco, CA, USA (2024)
9. Alam, M., Bhattacharya, S., Dutta, S., Sinha, S., Mukhopadhyay, D., Chattopadhyay, A.: Ratafia: Ransomware analysis using time and frequency informed autoencoders. In: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 218–227. IEEE, McLean, VA, USA (2019)
10. Woralert, C., Liu, C., Blasingame, Z.: Hard-lite: A lightweight hardware anomaly realtime detection framework targeting ransomware. IEEE Transactions on Circuits and Systems I: Regular Papers 70(12), 5036–5047 (2023)

11. Aurangzeb, S., Rais, R.N.B., Aleem, M., Islam, M.A., Iqbal, M.A.: On the classification of microsoft-windows ransomware using hardware profile. PeerJ Computer Science 7, 361 (2021)
12. Ganfure, G.O., Wu, C.-F., Chang, Y.-H., Shih, W.-K.: Deepware: Imaging performance counters with deep learning to detect ransomware. IEEE Transactions on Computers 72(3), 600–613 (2022)
13. Anand, P.M., Charan, P.S., Shukla, S.K.: Hiper-early detection of a ransomware attack using hardware performance counters. Digital Threats: Research and Practice 4(3), 1–24 (2023)
14. Babu, D.B., Nikhitha, P.S., Senthil, M., Babu, K.K.: Preventing ransomware attacks using host-based monitoring of processor and disk activity. In: Proceedings of International Conference on Computer Science and Communication Engineering (ICCSCE 2025), pp. 634–646. Atlantis Press, Hyderabad, India (2025)
15. Pan, Z., Shu, Z.: Hardware-assisted ransomware detection using automated machine learning. In: 2025 Design, Automation & Test in Europe Conference (DATE), pp. 1–7. IEEE, Lyon, France (2025)
16. Amira, A., Derhab, A., Karbab, E.B., Nouali, O.: A survey of malware analysis using community detection algorithms. ACM Computing Surveys 56(2), 1–29 (2023)
17. Regano, L., Canavese, D., Basile, C., Torchiano, M.: Empirical assessment of the code comprehension effort needed to attack programs protected with obfuscation. Computers & Security 166, 104881 (2026)
18. Singh, N., Tripathy, S.: Unveiling the veiled: An early stage detection of fileless malware. Computers & Security 150, 104231 (2025)
19. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 42(8), 2011–2023 (2020)
20. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? Advances in neural information processing systems 35, 507–520 (2022)
21. Wang, S., Dong, F., Yang, H., Xu, J., Wang, H.: Cancal: Towards real-time and lightweight ransomware detection and response in industrial environments. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pp. 2326–2340. Association for Computing Machinery, Salt Lake City, UT, USA (2024)
22. Lemmou, Y., Lanet, J.-L., Souidi, E.M.: In-depth analysis of ransom note files. Computers 10(11), 145 (2021)
23. ThreatLabz: An archive of ransomware notes past and present. https://github.com/ThreatLabz/ransomware notes, last accessed 2025/08/13
24. Sekar, A., Kulkarni, S.G., Kuri, J.: Learn: Leveraging ebpf and ai for ransomware nose out. In: 2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS), pp. 1323–1328. IEEE, Bengaluru, India (2025)
25. VirusShare.com: VirusShare.com - Because Sharing is Caring. https://virusshare.com, last accessed 2025/06/16
26. Abuse.ch: MalwareBazaar. https://bazaar.abuse.ch, last accessed 2025/06/16
27. CrowdStrike: Free Automated Malware Analysis Service - powered by Falcon Sandbox. https://hybrid-analysis.com, last accessed 2025/06/16
28. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research 20(177), 1–81 (2019)