Review of "SRDC: Semantics-based Ransomware Detection and Classification with LLM-assisted Pre-training"

Alex Towell
CS 590 - Graduate AI

May 2, 2025

1 Paper Information

Title: SRDC: Semantics-based Ransomware Detection and Classification with LLM-assisted Pre-training
Authors: Ce Zhou[1], Yilun Liu[2], Weibin Meng[2], Shimin Tao[2], Weinan Tian[2], Feiyu Yao[2], Xiaochun Li[2], Tao Han[2], Boxing Chen[3], Hao Yang[2]
Affiliations: [1]Northeastern University, China; [2]Huawei, China; [3]Huawei Canada, Canada
Venue: Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)
Year: 2025

2 Summary

Ransomware remains one of the most disruptive forms of malware, encrypting data and extorting payment for decryption keys. Traditional defenses typically rely on signature matching, static file features, or shallow machine-learning models trained on previously seen ransomware families. These approaches suffer two chronic weaknesses: (i) they fail against *zero-day* variants whose behavior differs from known samples, and (ii) they rarely leverage the *semantics* embedded in malware behavior or in expert domain knowledge.

The paper under review tackles both challenges by asking: *Can a detector that explicitly reasons over the semantics of malware behavior—and that is pre-trained with external expert knowledge—outperform current state-of-the-art systems, especially on unseen ransomware families?* To answer this, the authors introduce SRDC (Semantics-based Ransomware Detection and family Classification). The framework operates on dynamic execution traces (API calls, registry accesses, file operations, etc.) captured in a sandbox environment using the methodology established by Sgandurra et al. [1]. First, a *Feature Internal Semantic Processing* module rewrites terse identifiers such as `GetSystemInfo` into natural language phrases ("get system information"), enriching the raw trace with human-readable meaning. Second, a novel LATAP (LLM-assisted Task-Adaptive Pre-training) stage leverages a curated corpus of Windows API descriptions, registry semantics, and expert ransomware analyses—much of it generated and refined with a large language model—to adapt a compact GPT-2 model [2] toward the ransomware domain.

After LATAP, the GPT-2 backbone (termed *SRDC-GPT*) is fine-tuned on two tasks: (1) *ransomware family classification* and (2) *zero-day ransomware detection* (binary classification of unseen families versus benign software). The central hypothesis is that embedding both *internal semantics* (interpretable behaviors) and *external semantics* (expert knowledge) into the model will

yield better generalization and require far fewer labeled samples. SRDC's performance is compared against strong baselines, including BERT-based detectors, convolutional neural networks fed with raw traces, and recent ensemble techniques designed for zero-day scenarios.

3 Methodology

The SRDC framework employs a multi-stage approach to ransomware detection and classification. The methodology consists of three primary components:

3.1 Data Collection and Preprocessing

The authors gather dynamic execution traces by running ransomware samples in a controlled sandbox environment. These traces capture API calls, registry modifications, file system operations, and network activities—behavioral fingerprints that reveal malicious intent regardless of code obfuscation. The dataset, constructed by Sgandurra et al. [1], contains 582 ransomware samples across 11 families and 942 goodware (benign) instances. Each sample was executed for 30 seconds in the sandbox to collect behavioral traces. This balanced composition ensures robust model training and evaluation across diverse malware types.

3.2 Semantic Feature Engineering

The novel Feature Internal Semantic Processing module transforms low-level system calls into natural language descriptors. This transformation involves three specific techniques:

1. *Removing Special Abbreviations* - Windows API functions often include "W" or "A" suffixes (e.g., `OpenServiceW`, `WriteConsoleA`) denoting wide or ANSI character encoding. These suffixes are removed since they don't impact functionality but introduce inconsistent semantics.

2. *Converting Camel-cased Function Names to Verb Phrases* - API functions in CamelCase format are transformed into natural language phrases. For example, `GetFileSize` becomes "get file size" and `GetNativeSystemInfo` becomes "get native system information," making the behavior more explicit.

3. *Enhancing Features Semantically* - Non-standard syntax in feature records is converted to "verb + subject" structure. For instance, `REG:OPENED:HKEY_LOCAL_MACHINE\SYSTEM` becomes "opened registry HKEY_LOCAL_MACHINE\SYSTEM," creating a consistent and more interpretable representation.

This systematic transformation bridges the gap between machine-level activity logs and human-interpretable behaviors, enabling the model to reason over actions rather than mere syntactic patterns.

3.3 LLM-assisted Task-Adaptive Pre-training

Rather than starting from a generic language model, the authors leverage external knowledge through their LATAP approach. They first compile a specialized corpus containing Windows API documentation, cybersecurity reports on ransomware techniques, and expert analyses. This domain-specific corpus, enriched with LLM-generated examples, serves as pre-training material for a GPT-2 model [2], focusing its attention on security-relevant language patterns before

any supervised fine-tuning begins. This follows the principles of domain adaptation via continued pre-training established in previous work [3], though the authors present it as a novel approach specifically for the cybersecurity domain.

After these preparatory stages, the resulting SRDC-GPT model undergoes supervised fine-tuning using labeled ransomware samples for both binary classification (malicious vs. benign) and family classification tasks.

4 Key Findings

Semantic pre-training lowers language-model perplexity and boosts recall. After LATAP, SRDC-GPT's perplexity on unseen ransomware traces drops by 18.6 % relative to the base GPT-2, indicating deeper domain understanding. When fine-tuned for zero-day detection, this translates into a recall improvement of 3.19 % over the non-adapted model.

State-of-the-art accuracy on zero-day detection. On the benchmark dataset of Sgandurra et al. [1] the authors partition seven ransomware families (448 samples) for training and four families (134 samples) for testing, with an equal number of goodware samples in the test set. This experimental design deliberately simulates zero-day scenarios by ensuring family-level separation between training and testing sets. SRDC achieves 96 % accuracy and an F-score of 0.96 (Table 8), exceeding the previous best published method (DCAE-ZSLHVE) [4] which attains 93 % accuracy and 0.92 F-score. Recall reaches 97 %, critical for minimizing false negatives in security contexts.

Superior family-level classification. In multi-class experiments across eleven ransomware families plus benign software, SRDC attains a balanced accuracy of 54.8 % (Table 6). This outperforms VarLenPSO [5] (48.9 %), BERT [6] (50.3 %), LSTM [7] (43.2 %), and KNN baseline models. Per-family analysis shows advantage in eight of eleven families; notably, SRDC identifies the rare PGPCODER family (four samples) where VarLenPSO mis-classifies 100 % of cases.

Few-shot robustness. Figure 3 evaluates training with progressively fewer families. Even with only **two** ransomware families ($\approx 35\%$ of original data), SRDC still surpasses 90% detection accuracy on nine unseen families—an order-of-magnitude data reduction compared to earlier systems. The vanilla GPT-2 detector collapses under the same constraint, underscoring LATAP's importance.

Ablation validates each module. Removing the Feature Internal Semantic Processor, the feature-compression block, or the hierarchical combinatorial model yields noticeable drops in both balanced accuracy and F-score (Table 9). This confirms that semantics infusion at *both* representation and pre-training stages is necessary to reach top performance.

In aggregate, the evidence strongly supports the authors' thesis: semantics-aware pre-training plus behavior reconstruction provide measurable and repeatable gains over purely statistical or signature-based approaches.

5 Scientific Significance and Future Directions

Importance of the topic. Ransomware's rapid evolution and high societal cost make dependable detection and triage essential. Yet zero-day families continually circumvent static signatures. SRDC's ability to generalize from limited data directly addresses this gap, offering a path toward *resilient* malware defense that does not depend on exhaustive prior labeling.

Conclusions and empirical support. The paper concludes that enriching detectors with both internal and external semantics significantly improves accuracy, recall, and data efficiency. The

empirical evidence—multiple benchmark tables, per-family breakdowns, and few-shot curves—robustly backs these claims.

Advancement over prior work. Previous NLP-inspired detectors mostly used language models as generic encoders. SRDC pioneers a *domain-adapted* LLM that is explicitly pre-trained on security corpora and coupled with feature-level semantic rewriting. This hybrid approach addresses two key limitations of existing methods: insufficient use of internal semantics in ransomware features and inadequate leveraging of external knowledge from security experts. By transforming cryptic API calls and registry operations into natural language representations, SRDC enables the model to better understand the intentions behind software behaviors and potential security risks—advancing the state of the art in semantic malware analysis.

Critical perspective. While the results are impressive, several aspects warrant scrutiny. The authors present LATAP as a novel methodology, yet it largely resembles established domain-adaptive pre-training techniques that have been applied in various fields [3, 8]. The terminology may overstate the novelty of what is essentially continued pre-training with domain-specific corpora. Additionally, it remains unclear how much of SRDC's performance advantage stems from the semantic feature engineering versus the model architecture itself—an ablation study comparing semantically-enhanced features fed into simpler classifiers would have strengthened causal attribution. The decision to use GPT-2 rather than more recent language models is justified primarily on practical grounds, but leaves open questions about whether the approach scales efficiently to larger, more capable models.

Open questions. The current study evaluates only Windows ransomware traces collected in a controlled sandbox. Real-world deployment will require handling noisy, partial logs, adversarial obfuscation, and other operating systems. The authors' hierarchical approach to feature processing helps mitigate token length limitations in language models, but scaling to more complex execution traces remains a challenge. Moreover, the reliance on GPT-2 (124 M parameters) raises the question of whether larger, instruction-tuned models—or continual online adaptation—would yield further gains or merely increase inference cost. Formal adversarial testing against mimicry attacks is also absent and would be critical for production use.

Broader applicability. The SRDC blueprint is transferable. Any security task where expert semantics exist—phishing email detection, ICS intrusion, Android malware—could benefit from LATAP-style adaptation. Beyond cybersecurity, the framework exemplifies how domain-specific corpora plus LLMs can bootstrap high-performance models in low-resource settings, an idea with relevance to medical text mining, legal reasoning, and other specialized NLP applications.

Next research step. A promising direction is to embed SRDC within an *adversarial reinforcement-learning* loop: train a generative ransomware agent against a semantics-aware defender, yielding a continually improving detection system and a principled measure of robustness. Such closed-loop experimentation would test SRDC's limits and guide the design of even more resilient cyber-defense agents.

References

[1] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," *arXiv preprint arXiv:1609.03020*, 2016.

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[3] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, “Don’t stop pretraining: Adapt language models to domains and tasks,” *arXiv preprint arXiv:2004.10964*, 2020.

[4] U. Zahoora, M. Rajarajan, Z. Pan, and A. Khan, “Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier,” *Applied Intelligence*, vol. 52, no. 12, pp. 13941–13960, 2022.

[5] B. Tran, B. Xue, and M. Zhang, “A new representation in PSO for discretization-based feature selection,” *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1733–1746, 2017.

[6] T.-L. Lin, H.-Y. Chang, Y.-Y. Chiang, S.-C. Lin, T.-Y. Yang, C.-J. Zhuang, W.-L. Tseng, and B.-H. Zhang, “Ransomware Detection by Distinguishing API Call Sequences through LSTM and BERT Models,” *The Computer Journal*, vol. 67, no. 2, pp. 632–641, 2024.

[7] S. Maniath, A. Ashok, P. Poornachandran, VG. Sujadevi, P. S. AU, and S. Jan, “Deep learning LSTM based ransomware detection,” *2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, pp. 442–446, 2017.

[8] Y. Liu, S. Tao, W. Meng, J. Wang, H. Yang, and Y. Jiang, “Multi-Source Log Parsing With Pre-Trained Domain Classifier,” *IEEE Transactions on Network and Service Management*, 2023.