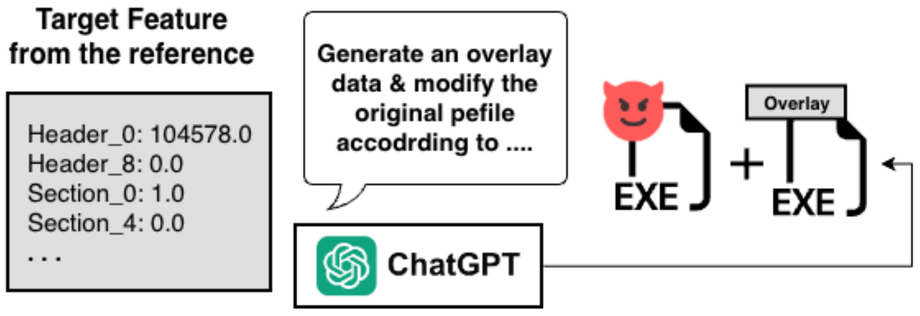
LLM Guided Adversarial Feature-to-Executable Malware Generation

Saeyeon Hong[1], Hyejin Woo[2], Md Mahmuduzzaman Kamol[3], Se Eun Oh[1], Mohammad Saidur Rahman[3]
[1]Ewha Womans University, Seoul, South Korea
[2]Dept. of AI Cyber Security, Korea University, Korea
[3]Dept. of Computer Science, University of Texas at El Paso, USA

Introduction

- VirusTotal [1] reports processing approximately 1.8 million samples every day.
- To remain effective in such environments, malware detection systems must not only be robust to drift but also resilient against adversarial manipulation.
- An adversarial malware sample is only a real threat if the perturbed binary can still execute while preserving its malicious functionality.
- Although RL-based methods [2] report improved performance, the learned policy does not explicitly guarantee evasive behavior.

Methods



In this work, we propose an LLM-guided feature-to-executable adversarial malware generation framework

- **Generating adversarial features with Shapley Additive Explanations (SHAP) :** We employ SHAP [3] to decompose each EMBER [4] sample (2,381 features) into additive contributions. To flip a prediction, we adopt a greedy perturbation strategy at each step, the feature with the largest positive contribution is replaced with a benign candidate value from a reference set that minimizes its SHAP score. The process repeats until the total contribution sum becomes negative, yielding a benign classification.

Algorithm 1: SHAP-Guided Greedy Perturbation

Input: Sample x, SHAP values $\{\phi_i(x)\}$, reference set R
Output: Perturbed sample x'
$x' \leftarrow x$;
while $\sum_i \phi_i(x') \geq 0$ **do**
  $j \leftarrow \arg\max_i \phi_i(x')$;
  // largest malicious contribution
  $v^\star \leftarrow \arg\min_{v \in R_j} \phi_j(x'_{[j]\leftarrow v})$;
  $x'_{[j]} \leftarrow v^\star$;
  recompute $\{\phi_i(x')\}$;
return x'

Algorithm 2: LLM-Guided Feature-to-Executable

Input: Perturbed features x', original PE b
Output: Executable $b^\star$
$b^\star \leftarrow b$;
prompt LLM to synthesize sections from x' and apply to $b^\star$;
prompt LLM with Δ_{byte} to generate overlay code;
apply overlay O_{byte} to $b^\star$;
return $b^\star$

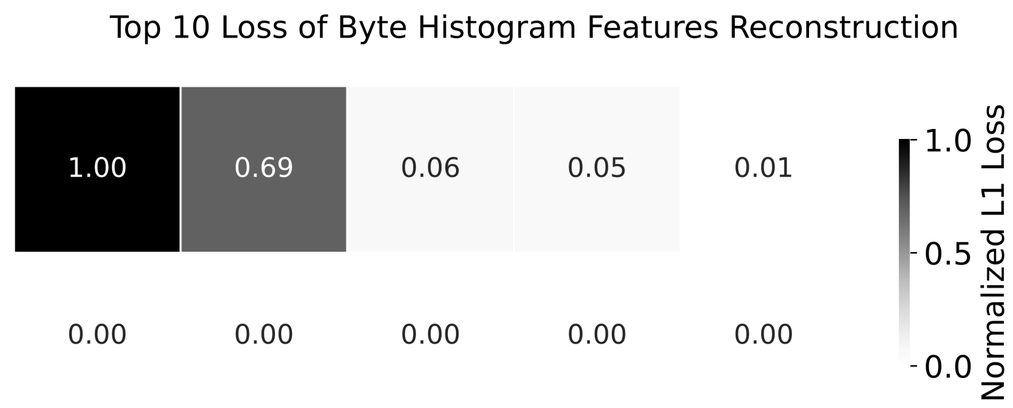
- **Feature-to-Executable Generation with Prompt:** We leverage LLMs to translate perturbed feature vectors into valid PE binaries. or byte-histogram features, the LLM is prompted to generate overlay code that adjusts byte counts without corrupting the file structure. For section level features, which often carry large SHAP contributions, the LLM synthesizes benign-looking payloads (e.g., ZIP, SQLite, BMP) with valid headers and alignment. This process enforces executability while approximating the adversarial target features.

> ***Prompt.*** *Please generate code that constructs the overlay Obyte so that the executable meets the following condition: feature j : target value $v^\star$.*

Evaluation

We evaluate our framework on 10,000 randomly selected malware samples from the VirusShare [5] dataset . The EMBER model is adopted as the target detector. We compare our method against MAB-malware [2], a reinforcement-learning based perturbation approach.

- **Attack Success Rate (ASR) :** Our method achieves a substantially higher ASR than MAB-malware (87.2% vs. 71.39%), demonstrating the effectiveness of SHAP-guided perturbations.
- **Executability**: we randomly selected 100 generated samples and submitted them to the Any.Run sandbox for dynamic analysis. All samples executed successfully.
- **Robustness of feature-to-executable translation**: While adversarial features achieved an ASR of 87.2%, the ASR measured on the corresponding binaries dropped to 55.14%. As shown in figure, most bytes are reconstructed with minimal error, but some target values deviate significantly when the required perturbation is too large to realize without excessive overhead. These reconstruction gaps directly reduce ASR when perturbations are translated into executable form.



Conclusion & Future Plans

We introduced an LLM-guided framework for feature-to-executable adversarial malware generation. Our approach combines SHAP-based feature selection with LLM-driven synthesis, achieving higher ASR than prior work while preserving executability. As future work, we aim to improve feature-to-binary fidelity and exploring methods to predict original feature representations from hashed features.

References

[1] VirusTotal. VirusTotal: Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community., 2025.
[2] Wei Song et al. MAB-Malware: A reinforcement learning framework for black-box generation of adversarial malware. In Proceedings of the ACM on Asia Conference on Computer and Communications Security (AsiaCCS), 2022.
[3] Scott M. Lundberg et al. From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2020.
[4] Hyrum S. Anderson et al. EMBER: an open dataset for training static PE malware machine learning models. arXiv preprint arXiv:1804.04637, 2018.
[5] VirusShare. VirusShare, 2025.