Can LLMs Handle WebShell Detection?

Overcoming Detection Challenges with a Behavioral Function-Aware Framework

Feijiang Han[1], Jianheng Tang[2], Yunhuai Liu[2], Jiaming Zhang[3], Chuyi Deng[3]

[1]University of Pennsylvania, [2]Peking University, [3]Central South University

TL;DR

- **The Problem:** Standard LLMs fail at WebShell detection. They get lost in irrelevant code (context limit) and learn from bad examples (ICL failure), leading to a crippling precision-recall trade-off.
- **Our Solution:** We built the BFAD framework that (1) surgically extracts only high-risk code and (2) selects in-context examples based on behavioral similarity, not just semantics.
- **The Impact:** BFAD enables LLMs to surpass state-of-the-art ML/DL methods without additional training. It dramatically boosts performance for all models (+13.82% avg), making large LLMs more accurate and small LLMs viable.

THE PROBLEM

A WebShell is a harmful script that serves as a backdoor on web servers, allowing remote access for attackers.

Traditional ML/DL detection methods struggle with new and obfuscated attacks due to their data demands. Although LLMs show promise, our systematic study reveals they have significant performance bottlenecks, hindering their ability to surpass current methods without further optimization.

Problem 1. Limited context cause LLMs to miss malicious code hidden in large files, while ineffective ICL examples mislead the model and waste precious space.

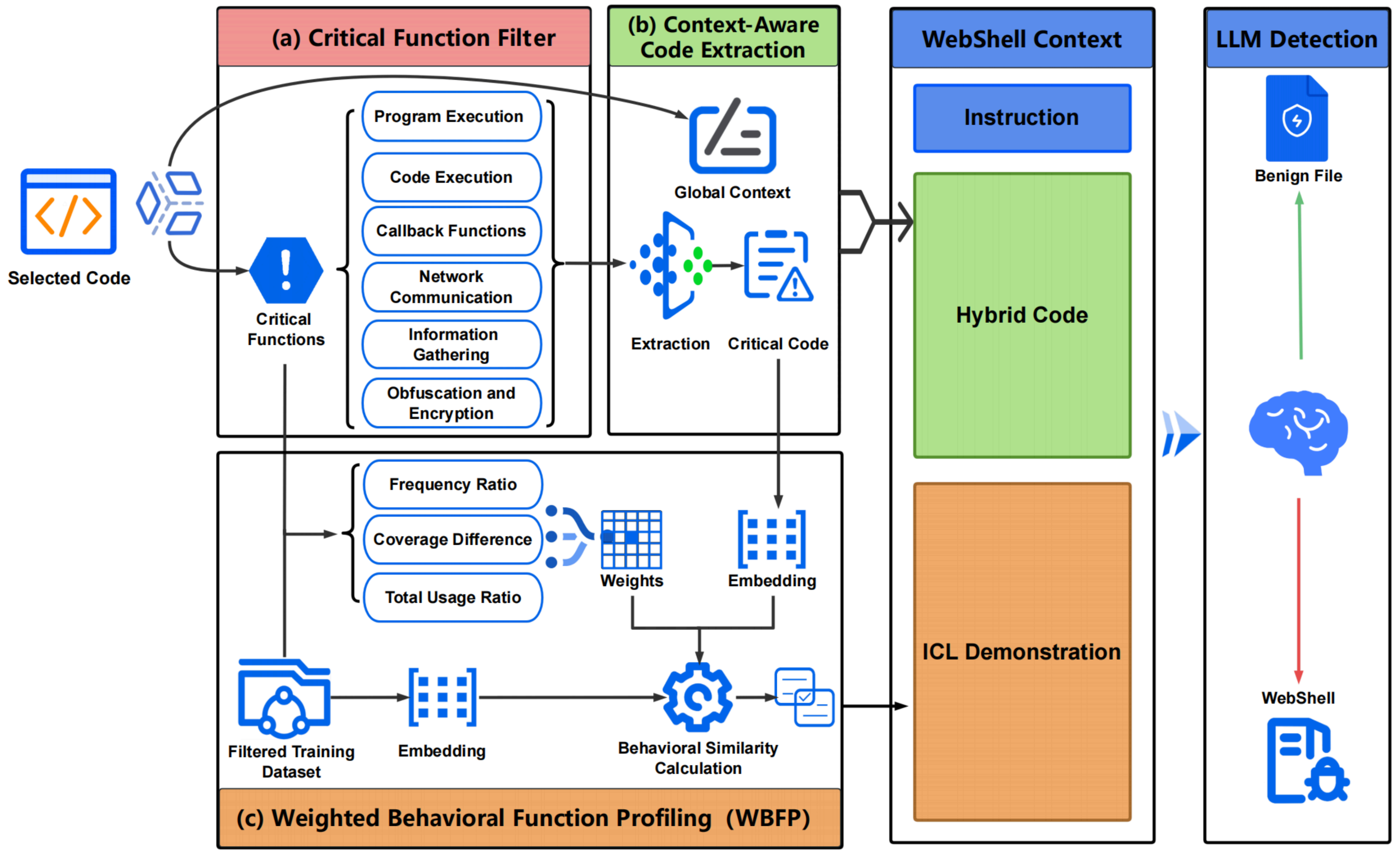
Problem 2. A Performance Dilemma:

- Large LLMs: Achieve high precision but miss real threats (Low Recall).
- Small LLMs: Suffer from a flood of false alarms (Low Precision) despite high recall.

OUR METHOD

Key Intuition:

To be effective, an LLM doesn't need to see more code; it needs to see the right code, guided by the right examples.



(a) Critical Function Filter, which identifies PHP functions associated with malicious behavior;
(b) Context-Aware Code Extraction, which isolates critical code regions to overcome LLM context limitations;
(c) Weighted Behavioral Function Profiling, which selects ICL demos using a behavior-weighted similarity score.

KEY RESULTS

Table 1: Performance Comparison of BFAD-Enhanced Models Against Baselines.

Category	Model	Accuracy	Precision	Recall	F1 Score
Sequence Baselines	GloVe+SVM	96.20%	93.30%	94.30%	93.80%
	CodeBERT+RF	96.30%	94.00%	95.60%	94.80%
Graph Baselines	GCN	96.90%	94.40%	95.30%	94.90%
	GAT	98.37%	99.52%	97.39%	98.87%
LLM Baselines (Large)	GPT-4	97.27%	100.00%	85.98%	92.46%
	LLaMA-3.1-70B	98.01%	97.31%	92.36%	94.77%
	Qwen-2.5-Coder-14B	98.64%	99.32%	93.63%	96.39%
LLM Baselines (Small)	Qwen-2.5-Coder-3B	71.11%	38.93%	99.32%	55.93%
	Qwen-2.5-3B	93.72%	78.03%	91.84%	84.37%
	Qwen-2.5-1.5B	43.62%	34.61%	95.77%	50.84%
	Qwen-2.5-0.5B	19.47%	18.65%	100.00%	31.44%
LLM + BFAD	GPT-4	99.75%	100.00%	98.71%	99.35% (+6.89)
	LLaMA-3.1-70B	99.38%	98.72%	98.09%	98.40% (+3.63)
	Qwen-2.5-Coder-14B	98.76%	98.68%	94.90%	96.75% (+0.36)
	Qwen-2.5-Coder-3B	78.89%	46.67%	100.00%	63.64% (+7.71)
	Qwen-2.5-3B	97.39%	88.64%	99.36%	93.69% (+9.32)
	Qwen-2.5-1.5B	80.40%	48.51%	100.00%	65.33% (+14.49)
	Qwen-2.5-0.5B	91.94%	71.10%	98.73%	82.67% (+51.23)

Our work proves that with the right guidance, even out-of-the-box, training-free LLMs can surpass traditional deep learning methods that require extensive training.

- **Surpassing SOTA:** Our BFAD-enhanced GPT-4 achieves a 99.35% F1 score, outperforming the training-based GAT model.
- **Making Small Models Viable:** BFAD dramatically improves Qwen-0.5B's F1 score by 51.23 points.
- **Universal Improvement:** Boosts average F1 score across all LLMs by 13.82%.