# LLM-driven Malware Analysis across Code Representations

Sil M. Folkertsma

## Introduction

- LLMs might be usable for Malware Detection.
- Possibly in Decompilation, possibly in Source Code Malware Detection.
- But what if it can already detect the Malware in Binary Code?

**RQ1: To what extent does the level of code representation influence the accuracy of LLM-driven malware analysis?**
New SBAN malware dataset as benchmark:
**RQ2: To what extent does the internal validity of the automated SBAN dataset support its reliability as a benchmark for LLM-based malware classification?**

## Methodology

RQ1:

- Source samples of malware and benign code across code representations.
- Prompt an LLM with a sample, telling it to analyze it and determine if it is malware or benign.
- Compare the output to original labels
- We aggregate the results by code representation.
- We evaluate the performance of the LLM on each code representation

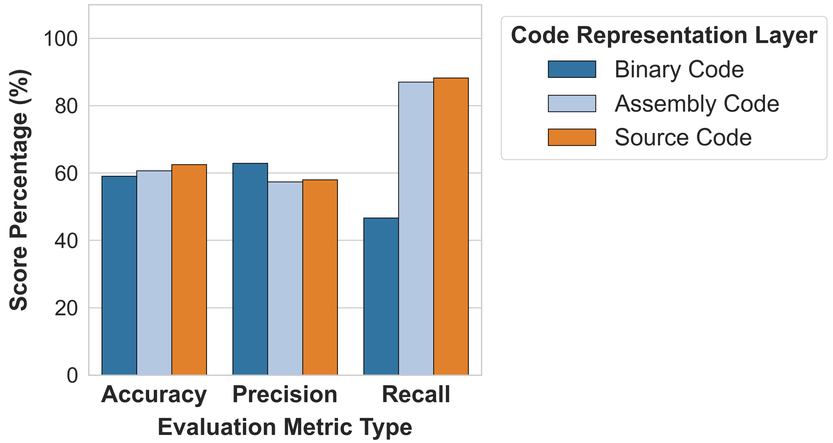
Performance: mainly recall; false negatives are very damaging.

RQ2:

- Clean the dataset (remove samples with missing fields)
- Sample from remaining samples
- Prompt LLM with sample, telling it to evaluate the correctness of the original label based on available code
- Aggregate results

We only use correctly labelled samples (from RQ2) for RQ1 experiments.

## Results




Experiment across Code Representations: Recall on Binary Code Significantly Worse

## Responsible Research

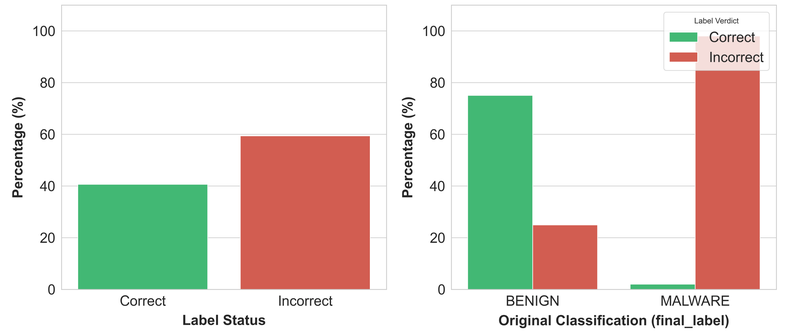
The main challenge in this research is ensuring reproducibility of LLM experiments:

- Same prompt for all representations
- Experiment with different prompt led to same conclusion
- Each experiment repeated 3 times
- LLM-as-a-judge ran 3 times with 99% consistency
- Used open-source model in local environment
- Ran with low temperature to ensure consistency

## Discussion

There were some limitations we encountered in this study:

- SBAN dataset was highly faulty -> small sample size
- Small sample size -> cannot tune prompt due to overfitting
- Small DelftBlue partition -> small model size
- Small model size -> greater distance to state-of-the-art, (presumably) lower accuracy




SBAN Dataset Audit Report: Most Malware Samples are Mislabeled

## Conclusion and Future Research

- SBAN is a highly inefficient benchmark
- LLMs perform worse on binary than other code representations

Future research should:

- Construct a more efficient benchmark
- Evaluate **why** binary performs worse
- Continue with LLM for decompilation
- Continue with LLM for high-level code malware detection
- Assess whether LLMs are inherently unusable in cybersecurity

S.M.Folkertsma@student.tudelft.nl