The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

How Effective are Large Language Models at Detecting Security Vulnerabilities?

Stuart D. Tioniwar (author)[1]

[1]Cambridge Centre for International Research

Abstract

With the prevalence of software systems and digitized information, (cyber) security vulnerabilities have also become an important focus. Large Language Models (LLMs) have been seen as potential agents in detecting these vulnerabilities. However, the question of "how effective can LLMs be in this role?" arises. This paper contributes a detailed evaluation of the LLMs' capabilities to carry out vulnerability detection (VD) along with the engineering of *OverallPrompt (OP)* and *OverallPrompt-Few Shot (OP-FS)*, as well as the *FunctionOnlyDataset* in an attempt to improve the LLMs' performance. This evaluation is measured by multiple metrics, including accuracy, F1, and *Bias rate*; focusing on quantitative binary results. Through meticulous analysis, this research paints a clear picture that LLMs are not suitable for real-world use as currently constructed: the best performances peak at 0.57 accuracy and 0.67 F1 score. This unreliability is not good enough for real-life situations where incorrect VD detections can result in catastrophic consequences. However, the improvement

in results when met with different prompts shows great promise in the capabilities of LLMs to further improve and result in more accurate detections by altering certain moving parts of the LLM network. Using various prompting, pretraining, and fine-tuning methods in the future can greatly impact the VD capabilities of LLMs and bring them further towards real-world reliability.

Keywords: LLM (Large Language Model), vulnerability detection, prompts, performance

Introduction

This era's transition into digitalization has brought efficiency to various systems, bringing information and data at the access of our very fingertips. However, with computerized systems comes the need to protect them from cyberterrorism and empower cybersecurity defenses.

In order to secure information systems, it is crucial to ensure that system codes are not susceptible/vulnerable to Security Vulnerabilities. Tools that have shown promise in this field are Large Language Models (LLMs). LLMs have shown great potential in various fields: security, code generation, healthcare, education, etc. (Fang et al., 2024) In particular, in the field of (cyber) security, multiple studies have been conducted that cite the potential of LLMs to detect vulnerabilities.

However, previous literature and studies have established the fact that LLMs are simply unfit for VD in real-world situations as currently constructed (Zhou et al., 2024). Previous

findings point to the lack of reasoning ability, robustness (Ullah et al., 2024), and LLM size (Zhou et al., 2024) as some of the main reasons attributed to poor performance in VD.

Given poor results in the past, this paper also engineered prompts, specifically *OP* and *OP-FS* to attempt to improve the performance of LLMs. These prompts utilize a combination of prompting techniques, specifically role prompting, structural instructions, chain-of-thought, and specific-CWE definition (Ullah et al., 2024) (Schullof et al., 2024) on top of data flow analysis (Khare et al., 2024). Furthermore, the OP-FS in particular is a few-shot prompt and hence contains exemplar vulnerable and patched code in its prompt.

In addition, this paper produced the FunctionOnlyDataset, a subset of patched and vulnerable code at the function level deduced from the SecLLMHolmes real-world datasets (Ullah et al., 2024); both contain a 50-50 split on the number of patched and vulnerable codes. In conducting VD testing, this paper aims to tackle five main questions, specifically:

RQ1. Can the results of previous experiments be replicated?

RQ2. Will function-level testing datasets produce better results than class-level code?

RQ3. Does the use of exemplars cause the results of VD testing to differ? (few-shot prompting VS. zero-shot prompting)

RQ4. How does the use of different models affect the VD testing results? RQ5. Which prompt is the most effective for conducting VD?

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Methodology

The VD testing is conducted on 13 prompts, mainly SecLLMHolmes' D1, D2, D3, D4 and D5 prompts (Ullah et al., 2024); our interpretation of those prompts, SD1-SD5; CWE-DF (Khare et al., 2024), which is a source-to-sink dataflow prompt; and OP and OP-FS.

Prompts are divided into 2 components: user-side prompts and system-side prompts. User-side prompts are used for the main instructions, mainly the testing code and direct questions (”does this code have a vulnerability?”). On the other hand, system-side prompts are used for additional information, such as structural instructions, exemplars, definitions, etc.

Few-shot prompts use vulnerable and patched code exemplars from the *SecLLMHolmes hand-crafted dataset* along with an explanation of why the code is either vulnerable or patched.

These 13 prompts are run across 2 models: *Llama3-8b-8192* and *Gemma2-9b-it*; as well as across 2 testing datasets: *SecLLMHolmes Dataset* and the *FunctionOnlyDataset*. Hence, four sets of data are achieved: (1) Llama3-8b-8192 with the SecLLMHolmes Dataset, (2) Llama3-8b-8192 with the FunctionOnlyDataset, (3) Gemma2-9b-it with the SecLLMHolmes Dataset, and (4) Gemma2-9b-it with the FunctionOnlyDataset.

Metrics

The resulting responses of the LLM VD testing are then stored and quantified into 4 base classifications: False Positives (FP), True Positives (TP), False Negatives (FN), and True Negatives (TN). These are then used to calculate for 5 quantifiable metrics: Accuracy, Recall,

Precision, F1 score and Bias.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Samples}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Bias = \frac{True\ Positives + False\ Positives}{Total\ Samples} - 0.5$$

Accuracy and F1 will be used as the main comparison metrics to gauge the effectiveness of LLMs based on VD tests, where F1 will be used as an overall (more effective evaluation) metric since it gauges that True Positives are more significant and important than True Negatives. On the other hand, the bias metric will be used to gauge possible one-sided results (biases) between prompts, datasets, and models: negative bias expresses results tilted to negative results, positive bias shows results tilted to positive results, and zero bias represents balanced results; as results from both positive and negative spectrums approach zero, results show more balance. This will be expressed through absolute value Bias rates.

Results

This section aims to present and compare the quantitative results across all experiments. Each prompt runs through 30 different patched and vulnerable codes for VD using the Groq API models' (Llama3-8b-8192, Gemma2-9b-it) default temperature settings. All raw data that have

been collected and discussed in this section are available here.

Summary of Data

Llama3-8b-8192

This part of the paper will focus on the results of the experiments carried out using the Llama3-8b-8192 model.

Table 1

FunctionOnlyDataset

	FP	FN	TP	TN	Accuracy	Recall	Precision	F1	Bias
D1	0	15	0	15	0.5	-	0	-	-0.5
D2	7	9	6	8	0.47	0.46	0.4	0.43	-0.07
D3	2	15	0	13	0.43	0	0	-	-0.43
D4	1	15	0	14	0.47	0	0	-	-0.47
D5	0	15	0	15	0.5	-	0	-	-0.5
SD1	4	12	3	11	0.47	0.43	0.2	0.27	-0.27
SD2	3	13	2	12	0.47	0.4	0.13	0.2	-0.33

SD3	0	15	0	15	0.5	-	0	-	-0.5
SD4	0	15	0	15	0.5	-	0	-	-0.5
SD5	0	15	0	15	0.5	-	0	-	-0.5
CWE-DF	13	1	14	2	0.53	0.52	0.93	0.67	0.4
OP	15	1	14	0	0.47	0.48	0.93	0.63	0.47
OP-FS	13	1	14	2	0.53	0.52	0.93	0.67	0.4

The accuracy across all prompts when run through the FunctionOnlyDataset remains consistent at a range of 0.47 to 0.5. However, there is an inconsistency in F1 scores across prompts: CWE-DF and OP-FS share the highest F1 scores with 0.67 while SD2 has the lowest score with 0.2.

SecLLMHolmes Prompts. Seven out of the ten SecLLMHolmes Prompts (D1-D5 and SD1-SD5) have 0 True Positives, leading to invalid (no) F1 scores. Looking at Bias rates, all SecLLMHolmes prompts have negative bias: wherein D2 has the lowest (absolute value) negative bias of -0.07, and D1, D5, SD3, SD4, and SD5 having completely lopsided results with a -0.5 bias. This shows one-sided results towards negative responses, which leads to its inability to be used in real-world situations, as vulnerabilities that do exist in the code will be left undetected and unattended to: vulnerable to attacks.

CWE-DF, OP, OP-FS. On the other hand, the prompts that contain data flow, mainly CWE-DF, OP, and OP-FS have higher F1 scores of 0.63 to 0.67. However, this is a result of (contrary to SecLLMHolmes prompts) their positively skewed bias rates of 0.4, 0.47, and 0.4. This high positive bias due to the prompt and hence higher True Positive rate causes F1 scores to increase by a substantial amount compared to those of the SecLLMHolmes prompts despite similar accuracy rates. This high Positive Bias, although greatly reduces possibilities of unchecked vulnerabilities, will cause both time and monetary inefficiency in real-world settings due to both time and funding requirements in analyzing False Positives.

Table 2

SecLLMHolmes Dataset

	FP	**FN**	**TP**	**TN**	**Accuracy**	**Recall**	**Precision**	**F1**	**Bias**
D1	2	14	1	13	0.47	0.33	0.67	0.11	-0.4
D2	10	5	10	5	0.5	0.5	0.67	0.57	0.17
D3	1	15	0	14	0.47	0	0	-	-0.47
D4	2	14	1	13	0.47	0.33	0.07	0.11	-0.4
D5	1	15	0	14	0.47	0	0	-	-0.47
SD1	7	10	5	8	0.43	0.42	0.33	0.37	-0.1

SD2	7	11	4	8	0.4	0.36	0.27	0.31	-0.13
SD3	3	12	3	12	0.5	0.5	0.2	0.29	-0.3
SD4	4	13	2	11	0.43	0.33	0.13	0.19	-0.3
SD5	4	13	2	11	0.43	0.33	0.13	0.19	-0.3
CWE-DF	14	1	14	1	0.5	0.5	0.93	0.65	0.43
OP	15	0	15	0	0.5	0.5	1	0.67	0.5
OP-FS	15	0	15	0	0.5	0.5	1	0.67	0.5

Accuracy across all prompts when run across the SecLLMHolmes Dataset remains consistent at a similar range as that of in Table 1 with a 0.43-0.5 range. However, the number of True Positives increased across all SecLLMHolmes Prompts, causing an increase in the F1 score across the board; whereas True Positives and F1 scores of the dataflow prompts remained at similar values.

SecLLMHolmes Prompts. Negative bias rates across SecLLMHolmes prompts have slightly approached 0 (lesser bias and an increased number of positives) with D3 and SD1 having the least (absolute value) bias of -0.1. Due to the additional context of the class-level code as opposed to the function-level code, Llama3-8b-8192 utilized additional textual and

contextual cues to determine that more code files had vulnerabilities.

CWE-DF, OP, OP-FS. On the other hand, F1, accuracy, and bias were consistent for dataflow prompts between the FunctionOnlyDataset and the SecLLMHolmes Dataset. ***Gemma2-9b-it***

This part of the paper will dive into the results of the experiments carried out using the Gemma2-9b-it model.

Table 3

FunctionOnlyDataset

	FP	**FN**	**TP**	**TN**	**Accuracy**	**Recall**	**Precision**	**F1**	**Bias**
D1	12	2	13	3	0.53	0.52	0.87	0.65	0.33
D2	10	4	11	5	0.53	0.52	0.73	0.61	0.3
D3	7	8	7	8	0.5	0.5	0.47	0.48	-0.03
D4	6	8	7	9	0.53	0.54	0.47	0.5	-0.07
D5	7	7	8	8	0.53	0.53	0.53	0.53	0
SD1	11	4	11	4	0.5	0.5	0.73	0.59	0.23

SD2	11	4	11	4	0.5	0.5	0.73	0.59	0.23
SD3	8	7	8	7	0.5	0.5	0.53	0.51	0.03
SD4	9	6	9	6	0.5	0.5	0.6	0.55	0.1
SD5	5	8	7	10	0.57	0.58	0.47	0.52	-0.1
CWE-DF	15	0	15	0	0.5	0.5	1	0.67	0.5
OP	13	2	13	2	0.5	0.5	0.87	0.64	0.37
OP-FS	12	3	12	3	0.5	0.5	0.8	0.62	0.3

The accuracy remained consistent across all prompts in a range of 0.5 to 0.57, with SD5 having the 4 highest accuracy. Furthermore, F1 scores are confined in a small range of 0.48 to 0.67 and are therefore consistent across all prompts.

SecLLMHolmes Prompts. (Absolute value) Bias rates in SecLLMHolmes prompts are lower than results from the Llama3-8b-8192 model, showing less bias: with D5 having a completely balanced result of 0 and the rest ranging from -0.07 to 0.33 (showing relatively balanced bias rates).

CWE-DF, OP, OP-FS. On the other hand, Dataflow prompts (still) have high positive bias, meaning there are high amounts of positives and low amounts of negatives.

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Table 4

SecLLMHolmes Dataset

	FP	FN	TP	TN	Unintelligible	Accuracy	Recall	Precision	F1	Bias
D1	4	6	2	6	12	0.27	0.33	0.25	0.28	-0.3
D2	7	3	6	2	12	0.27	0.46	0.67	0.55	-0.07
D3	2	5	2	5	16	0.23	0.5	0.29	0.37	-0.37
D4	1	5	1	5	18	0.2	0.5	0.17	0.25	-0.43
D5	1	5	1	5	18	0.2	0.5	0.17	0.25	-0.43
SD1	3	6	3	7	11	0.33	0.5	0.33	0.4	-0.3
SD2	5	5	4	3	13	0.23	0.44	0.44	0.44	-0.2
SD3	1	6	1	6	16	0.23	0.5	0.14	0.22	-0.43
SD4	1	5	2	6	16	0.27	0.67	0.29	0.4	-0.4
SD5	0	6	1	6	17	0.23	1	0.14	0.25	-0.47
CWE-DF	7	2	6	0	15	0.2	0.46	0.75	0.57	-0.07
OP	8	1	8	1	12	0.3	0.5	0.89	0.64	0.03

OP-FS	4	3	4	3	16	0.23	0.5	0.57	0.53	-0.23

Table 4 introduces a new column: unintelligible. 'Unintelligible' results refer to LLM responses in which binary results cannot be extracted. This could refer to unclear responses or errors in execution. These unintelligible results were achieved as a result of longer prompts (including exemplars and testing code) and caused a decrease in both bias (to negative bias rates ranging from -0.07 to -0.47) and accuracy. Although the F1 results are not affected by unintelligible results, the negative bias shows low positive (and hence true positive) results and thus generally lower F1 scores ranging from 0.25 to 0.64.

Due to a high number of unintelligible results, the accuracy plummeted to a range of 0.2 to 0.33. Furthermore, despite being statistically unaffected by unitelligible results, F1 scores for SecLLMHolmes prompts drop to a range of 0.22 to 0.55. These over-arc Gemma2-9b-it's incompetence in conducting VD with larger (class-level code) inputs.

Taking away unintelligible results and focusing on properly-processed data may serve as a better basis of the model's capabilities with class-level code within the scope of its maximum input capacity that produces legible results.

Table 5

SecLLMHolmes Dataset without 'Unintelligible'

	Accuracy	**Bias**
D1	0.44	-0.17
D2	0.44	0.22
D3	0.5	-0.21
D4	0.5	-0.33
D5	0.5	-0.33
SD1	0.53	-0.18
SD2	0.41	0.03
SD3	0.5	-0.36
SD4	0.57	-0.29
SD5	0.54	-0.42
CWE-DF	0.4	0.37
OP	0.5	0.39

OP-FS	0.5	0.07

Taking into account legible results, the accuracy returns to the 0.4-0.5 range, which is a decrease from the results of the FunctionOnlyDataset where the accuracy range was 0.5-0.57.

SecLLMHolmes Prompts. However, a bias (decrease) toward negative results in SecLLMHolmes prompts is still visible in comparison to the results of the FunctionOnlyDataset with the exception of D2. As a result, with the exception of D2 and SD2, all results have a negative bias, with SD2 having the smallest bias of (positive) 0.03.

CWE-DF, OP, OP-FS. While the positive rates for OP remained constant (0.37 in the SecLLMHolmes data set to 0.39 in the FunctionOnlyDataset), CWE-DF faced a negative rate decrease of 0.13 while OP-FS faced a drop of 0.23: both substantial decreases.

Performance of OP and OP-FS Prompts

The OP and OP-FS prompts contain a combination of both the dataflow aspects and the SecLLMHolmes prompts aspects, mainly: definition and role prompting. However, despite a combination of the two, OP and OP-FS perform better head-to-head compared to SecLLMHolmes prompts and have very similar results to the CWE-DF prompt in terms of F1. Definition and role prompting are simply additional information given in the system-side prompts, whereas dataflow prompting are direct instructions to identify and track information from source to sink. Hence, the results are closely related to CWE-DF.

OP and OP-FS have very similar performances; OP barely beats out OP-FS with an average F1 score of 0.645 versus OP-FS's 0.6225. This difference can be greatly attributed to F1 scores in Gemma2-9b-it's performance in the SecLLMHolmes dataset wherein OP had 0.64 F1 in comparison to OP-FS's 0.53. This can be attributed to the additional length of Few-Shot prompting (due to exemplars) and Gemma's inability to handle longer prompts. Furthermore, both prompts have very similar biases in Table 1 (0.47 vs 0.4), Table 2 (0.5 vs 0.5), and Table 3 (0.37 vs. 0.3). However, in Tables 4 and 5, they differ greatly (0.03 vs -0.23 and 0.39 vs 0.07). The inability of the Gemma model to handle larger-sized prompts caused uncertainty when prompted with OP-FS, causing an increase in negative results for OP-FS as opposed to OP.

RQ1: Replication

This experiment replicates both SecLLMHolmes D1-D5 Prompts (Ullah et al., 2024) and the CWE-DF Prompt (Khare et al., 2024). This subsection aims to investigate whether results of experiments done by these past studies can be replicated - and to what extent.

SecLLMHolmes Replication

The original SecLLMHolmes paper mentions that Few-Shot prompts yield higher accuracy than No-Shot ones (Ullah et al., 2024). However, our results show that accuracy is consistent throughout different prompts, whether few-shot or no-shot. Furthermore, the original paper states that LLMs' reasoning abilities have much to improve on. This is backed up through a read-through of prompting results wherein quite a number of LLM explanation point to wrong aspects of the code as containing a vulnerability-regardless of binary result.

CWE-DF Replication

Khare, A., et al. maintained in "Understanding the Effectiveness of Large Language Models in Detecting Security Vulnerabilities" that GPT-4's best performances with the CWE-DF model had F1 scores of 0.69, 0.76, and 0.70. Our experiments show that the 3 best CWE-DF performances had F1 scores of 0.67, 0.67, and 0.64, which are comparable to the paper's results despite the use of smaller models (than GPT-4).

RQ2: Input Data Granularity

This subsection aims to tackle how additional context will affect the LLM's performance.

FunctionOnlyDataset contains function-level testing sets as opposed to class-level testing sets in the SecLLMHolmes Dataset. Class-level sets contain more code context and hence more contextual and textual cues than function-level sets.

The use of the different datasets mainly influences SecLLMHolmes Prompts' results, while CWE-DF, OP, and OP-FS retain similar results. The number of positive results generally increases in the Llama model when using the SecLLMHolmes Dataset as opposed to FunctionOnlyDatasets. Proportionally, bias rate also generally improves (the difference from 0 decreases). However, accuracy remains relatively identical; while the SecLLMHolmes dataset increases F1 scores in SecLLMHolmes prompts, generally meaning that performance of the Llama model is better when run through the SecLLMHolmes dataset.

This means that the additional context of a class-level set allows the Llama model to use

textual and contextual cues to detect more True Positives (hence the increased positive rate) and thus increases F1 scores aggregately.

As previously established in Section 3.1.2, the Gemma model is not best suited for larger input sizes and therefore does not perform well in the SecLLMHolmes dataset. Even in those that it is able to produce legible results, the accuracy rates decrease compared to results when tested against the FunctionOnlyDataset. Furthermore, F1 scores also decrease drastically in SecLLMHolmes prompts, while the decrease is not as drastic in dataflow prompts.

The additional context of class-level code seems to overwhelm the Gemma model as it is unable to detect vulnerabilities effectively, wherein the uncertainty caused by additional context generates (standard) negative results; thus, a general decrease in number of positive results (leading to generally more negative biases) in addition to its lower F1 scores. Hence, this model is not well-suited for larger input granularities (class level) and it may be optimal to use in function-level or line-level VD.

RQ3: Few-Shot VS. Zero-Shot

Table 6

Few-Shot and Zero-Shot Prompts

Few-Shot	**Zero-Shot**
D3/SD3	D1/SD1
D4/SD4	D2/SD2
D5/SD5	CWE-DF
OP-FS	OP

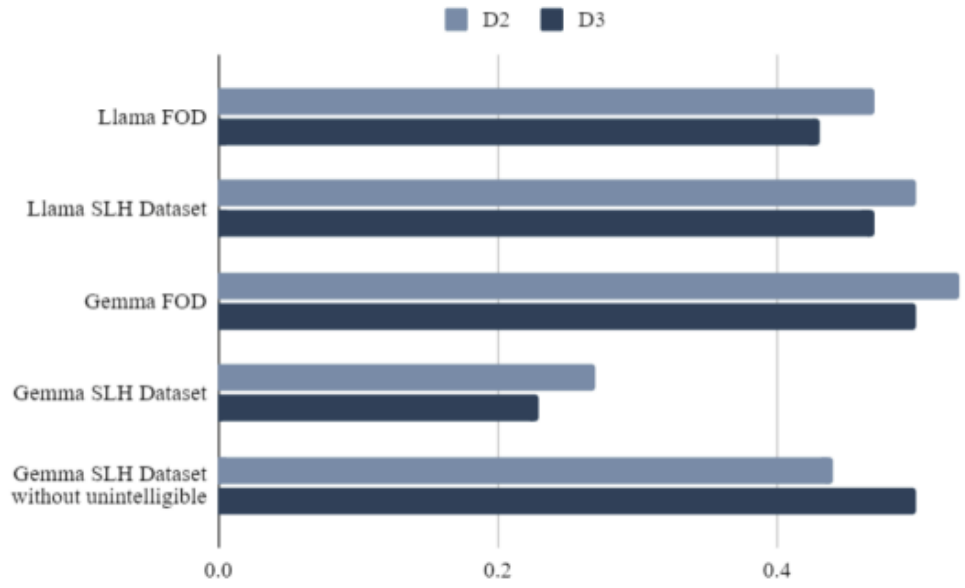
This subsection aims to answer the question: How will giving exemplars as part of the prompts affect the capabilities of the LLMs to conduct VD? The basis of comparison will be held between D2/SD2 against D3/SD3 and OP against OP-FS as the two sides are identical with a singular difference: the presence of exemplars.

This subsection will utilize accuracy as the medium of comparison instead of F1 as some results do not produce an F1 result due to their absence of True Positives. However, F1 scores will still hold some weight in the comparison leading to a definitive conclusion.

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Figure 1

D2 VS. D3 Accuracy Rates

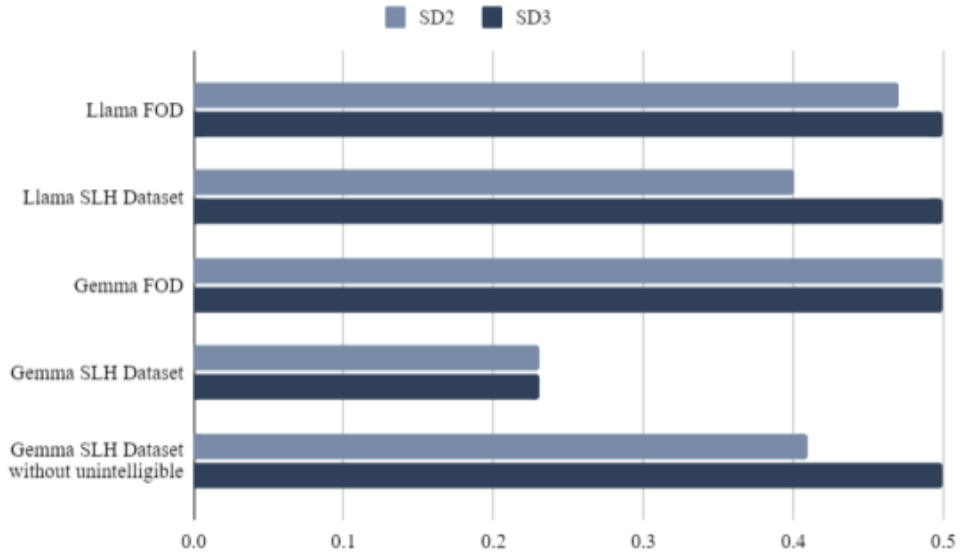


D2 VS. D3. All 4 main experiments had D2 performing slightly better in terms of accuracy compared to D3, and their F1 scores tell the same story. Interestingly, however, D3 performs better than D2 when unintelligible results are taken out of the equation (Gemma - SecLLMHolmes Dataset). However, D2's F1 score still outperforms D3's in this situation 0.55 to 0.37 (F1 is a metric unaffected by unintelligible results).

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Figure 2

SD2 VS. SD3 Accuracy Rates



SD2 VS. SD3. SD3 performs better than SD2 in general as it has a higher accuracy rate in both Llama experiments while they have equal accuracy rates in both Gemma experiments. However, when removing unintelligible results from the Gemma SecLLMHolmes Dataset results, the trend shown in Llama holds true as SD3 again outperforms SD2 in accuracy.

However, F1 scores for SD2 and SD3 do not support this trend as in all applicable F1 scores (experiments where there is at least 1 TP), SD2 beats out SD3; as an overall metric, F1 hence concludes that SD2 actually performs better than SD3 for its ability to attain True Positives.

OP VS. OP-FS. As stated in Section 3,2, OP and OP-FS have very similar performances

across all experiments. However, its difference in Gemma SecLLMHolmes Dataset's F1 results (where OP outperforms OP-FS) can be attributed to Gemma's inability to handle larger input granularities.

It can be concluded that few-shot and Zero-Shot prompting does not affect results in terms of Dataflow prompts. On the other hand, Zero-Shot prompting improves performance of LLMs as opposed to Fewshot prompting. Generally speaking, Zero-Shot prompting will produce better results; though it may differ according to the prompt.

RQ4: Model as a Factor to VD

The Gemma model is ineffective with longer prompts. Hence, its poor performance when run on the SecLLMHolmes dataset. However, it is capable of detecting True Positives relatively well when run against the FunctionOnlyDataset (as shown by the F1 scores). The Gemma model is more effective than its Llama model counterpart in conducting VD with smaller input sizes (perhaps function-level or line-level inputs). On the other hand, the Llama model is more effective in larger input sizes (class-level inputs) than the Gemma model.

Furthermore, different models respond differently to additional context (as established in Section 3.4). However, another notable difference is the different models' responses to lack of context - mainly when run through the FunctionOnlyDataset.

When the models face a lack of code context, dataflow prompts consistently produce high positive bias, meaning that positive results are dominant. However, SecLLMHolmes

Prompts allow its models to respond differently. When the Llama model is run on the SecLLMHolmes Prompts through the FunctionOnlyDataset, it responds with low positive rates; in contrast, the Gemma model responds with high positive rates.

RQ5: Prompt Comparison

This part of the paper aims to compare all 13 prompts used in the conduct of the experiments to find the most effective one. This will take into account average accuracy rates, average F1 scores, and average bias. Non-results (F1 scores with a ”-” result) will not be considered and removed completely from the equation. Furthermore, accuracy and bias results from the Gemma model tests on the SecLLMHolmes dataset will be taken from Table 5 as opposed to table 4.

Table 7

Average VD Results

	Accuracy	**F1**	**Bias**
D1	0.485	0.35	-0.185
D2	0.485	0.54	0.13
D3	0.475	0.425	-0.285
D4	0.4925	0.29	-0.3175

D5	0.5	0.39	-0.325
SD1	0.4825	0.4075	-0.08
SD2	0.445	0.385	-0.05
SD3	0.5	0.34	-0.2825
SD4	0.5	0.38	-0.2475
SD5	0.51	0.32	-0.33
CWE-DF	0.4825	0.64	0.425
OP	0.4925	0.645	0.4325
OP-FS	0.5075	0.6225	0.3175

Individual Criteria Evaluation

The accuracy for all are relatively similar, ranging from 0.4825 to 0.51. With SD5 having the (slightly) highest accuracy of 0.51. F1 scores and biases, on the other hand, vary greatly in comparison. F1 ranges from 0.29 (D4) to 0.645 (OP), while (absolute value) bias ranges from 0.05 (SD2) to 0.4325 (OP). With Dataflow prompts performing better than SecLLMHolmes prompts in terms of F1 but worse in terms of Bias. Hence, in comparing the prompts, a balance must be struck between the F1 score and Bias rate.

Comparative Evaluation

Taking a look at the top 3 F1 performers, CWE-DF, OP, and OP-FS are all Dataflow prompts. Unfortunately, the 3 dataflow prompts also share high positive biases (0.425, 0.4325, and 0.3175) as mentioned in Section 3.2. Thus, the high true positive count that results in high F1 scores could be due to this possible bias.

Taking a look at the remaining SecLLMHolmes prompts, SD1 and SD2 have the lowest absolute value biases with -0.08 and -0.05 respectively. However, their F1 scores of 0.4075 and 0.385 are low. However, taking a look at the next lowest bias, the results present D2. With its fourth highest F1 (after the dataflow prompts) of 0.54 and its third lowest (absolute value) bias of 0.13, D2 has the most outstanding performance out of all the prompts (both the SecLLMHolmes and the Dataflow prompts).

Related Work

This paper was inspired by multiple previous literature. This section aims to cover how other literature conducted VD through 3 main highlights: Prompting Techniques, VD Methodologies, and Underwhelming Performance in VD Testing.

Prompting Techniques

Creating effective prompts is extremely crucial. Using prompting techniques such as Role Prompting, Chain-of-Thought, Output Formatting (Schulhoff et al., 2024) allows for improved communication with LLMs and (theoretically) better results from the LLM. Some

other components of a prompt that contribute to LLM-client communication are: exemplar label quality, few-shot and no-shot prompting, and thought-articulation instructions.

These different prompting techniques aim to smoothen natural language structures and allow LLMs to more effectively convert Natural Language into computerized language that is processable. Hence, allowing for instructions to be followed directly and producing results that are favorable to users. In the case of this paper, Role Prompting, Chain-of-Thought, Output Formatting, Few-shot and no-shot prompts, additional context (definition of certain CWEs), dataflow prompting, and more were utilized.

LLM VD Methodologies

Past VD tests have been conducted to test the abilities of LLM to carry out VD. Earlier works have used different proposed frameworks, guidelines, and datasets such as SecLLMHolmes (Ullah et al., 2024), AutoCodeRover (Zhang et al., 2024), as well as PrimeVul and VD-S (Ding et al., 2024). Furthermore, different prompt templates (Schulhoff et al., 2024) were used to perform VD. The SecLLMHolmes framework maintains its use of 17 different prompt templates, utilizing role-prompting, zero-shot and few-shot prompting, etc. (Ullah et al., 2024) Whereas other recent works have utilized pair-wise testing (Risse et al., 2024), dataflow analysis (Khare et al., 2024), asking for explanation (Ullah et al., 2024) (Zhang et al., 2024) (Ding et al., 2024) (Khare et al., 2024), human-in-the-loop (Zhang et al., 2024), and CWE-specific VD (Ullah et al., 2024) (Khare et al., 2024).

However, in conducting VD testing, other works have taken different approaches. In

particular, changes to datasets: code augmentations (to test for robustness) (Ullah et al., 2024), 4, (Khare et al., 2024), code copies de-duplication, and chronological splitting (Ding et al., 2024). Using different methods to test LLMs' abilities to react to change or to find 'dataset-cleansing' methods that would optimize datasets to yield the best results in VD. An example of code augmentation is trivial changes in testing dataset codes (such as whitespaces or changes in variable names) (Ullah et al., 2024).

Recent works have also tested the effectiveness of different types of models in performing VD, mainly transformer-based models (Thapa et al., 2022) versus traditional neural networks. Transformer-based models, which can perform parallel processing; Static Analysis Tools (Khare et al., 2024), which analyze programs without executing the code; and deep learning tools (Khare et al., 2024) are stacked against each other to find which of the different LLM models is best suited for VD.

Lack of Correctness in LLM VD

Through analysis of LLM VD tests, conclusions have been reached that share the same overall result: LLMs yield inaccurate/underwhelming results when conducting VD. These conclusions are met with varying perspectives: lack of reasoning ability (Ullah et al., 2024) (Ding et al., 2024) (Risse & B¨ohme., 2024) (Khare et al., 2024), overreliance on textual cues compared to contextual understanding (Ding et al., 2024) (Risse & B¨ohme., 2024) (Khare et al., 2024), need for further context (Zhang et al., 2024) (Ding et al., 2024), and data leakage (Thapa et al., 2022).

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Conclusion

This paper finds that there are an array of different factors that may cause biases or "one-sided" data, as well as an improvement or a deterioration in LLM performance. Some of these factors include: choice of model, the use of exemplars, the size of prompts/input data, and the structure and techniques used in various prompts.

However, it is notable that although one model may be better at performing certain VD tasks than the other, both models have displayed poor overall performance, with no F1 score eclipsing the 0.7 mark (benchmark for what is considered to be "good"). Hence, showing that LLMs have a long way to go before being able to be confidently and effectively used in real-world settings to conduct Vulnerability Detection.

Despite this, this paper finds that LLMs have a high potential in conducting VD. Although this paper may display low results, the 2 models have shown promise by displaying the impact of certain LLMs along with certain factors that substantially improve VD results.

Limitations and Threats to Validity

Despite covering multiple angles of the issue, this paper narrows down to two specific LLM models, the gemma2-9b-it and the llama3-8b-8192 models. Furthermore, the paper also focuses mainly on 2 denominations of prompts: particularly SecLLMHolmes prompts (D1-D5 and SD1-SD5) and dataflow prompts (CWE-DF, OP, and OP-FS). Due to the small sample size of both LLM models and prompt types, the paper may not fully encompass results that will

reflect on all LLMs.

Furthermore, the gemma model's restrictive capabilities in handling larger prompts led to high amounts of 'unintelligible' results. This created a dent in the integrity of the paper's results; producing anomalistic changes in performance that may not properly represent LLMs' capabilities in conducting VD as a whole.

Future Direction

This paper has covered a multitude of VD faucets, mainly: the investigation into class-level testing, exploration of dataflow prompts, and few-shot prompting. However, this field still has an array of underexplored areas that could provide further insight on how to improve LLMs capabilities in conducting VD.

Future experiments and studies could use larger models, such as the GPT models, and again utilize large input granularities to further explore that area. Furthermore, Zhou, X., et al. also maintains that a possible limitation of previous VD studies has been the lack of LLM-workflow or LLM-user interactions. This can be incorporated into future studies in the form of user feedback on training set results to optimize testing set results. Additionally, previous studies have mentioned that certain LLMs may be better geared to detect certain CWEs (Khare et al., 2024). This can be explored by testing LLMs on specific CWEs as opposed to full datasets with a combination of CWEs.

Lastly, this paper takes into account quantitative data and does not fully cover the

qualitative aspect of the results. Hence, future studies should also review and analyze explanations to gauge whether LLMs are able to reason correctly and effectively.

The Princeton Journal of Interdisciplinary Research
ISSN 3069-8200

Acknowledgements

I would like to acknowledge *Professor Sergio Maffeis* from Imperial College London for having guided me throughout the research, analysis, literature review, and writing process of this paper. Thanks to his help, I have been able to gain better insight and improve upon the quality of this paper/report.

References

Ding, Y., Fu, Y., Ibrahim, O., Sitawarin, C., Chen, X., Alomair, B., Wagner, D., Ray, B., & Chen, Y. (2024). Vulnerability detection with code language models: How far are we? ArXiv. https://doi.org/10.48550/arXiv.2403.18624

Fang, C., Miao, N., Srivastav, S., Liu, J., Zhang, R., Fang, R., Asmita, Tsang, R., Nazari, N., Wang, H., & Homayoun, H. (2024). Large language models for code analysis: Do LLMs really do their job? In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24) (pp. 829–846). USENIX Association. https://www.usenix.org/conference/usenixsecurity24/presentation/fang

Khare, A., Dutta, S., Li, Z., Solko-Breslin, A., Alur, R., & Naik, M. (2023). Understanding the Effectiveness of Large Language Models in Detecting Security Vulnerabilities. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.2311.16169

Risse, N., & Böhme, M. (2024). Uncovering the limits of machine learning for automatic vulnerability detection. In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24) (pp. 4247–4264). USENIX Association. https://www.usenix.org/conference/usenixsecurity24/presentation/risse

Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., Schulhoff, S., Dulepet, P. S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., Da Costa, H., Gupta, S., Rogers, M. L., Goncearenco, I.,

Sarli, G., Galynker, I., Peskoff, D., Carpuat, M., White, J., Anadkat, S., Hoyle, A., & Resnik, P. (2024). The Prompt Report: A systematic survey of prompting techniques. arXiv. https://doi.org/10.48550/arXiv.2406.06608

Thapa, C., Jang, S. I., Ahmed, M. E., Camtepe, S., Pieprzyk, J., & Nepal, S. (2022). Transformer-Based Language Models for Software Vulnerability Detection. Proceedings of the 38th Annual Computer Security Applications Conference. https://doi.org/10.1145/3564625.3567985

Ullah, S., Han, M., Pujar, S., Pearce, H., Coskun, A., & Stringhini, G. (2024). LLMs cannot reliably identify and reason about security vulnerabilities (yet?): A comprehensive evaluation, framework, and benchmarks (2024 IEEE Symposium on Security and Privacy, pp. 862–880). IEEE. https://doi.org/10.1109/SP54263.2024.00210

Zhang, Y., Ruan, H., Fan, Z., & Roychoudhury, A. (2024). AutoCodeRover: Autonomous program improvement. ArXiv. https://doi.org/10.48550/arXiv.2404.05427

Zhou, X., Cao, S., Sun, X., & Lo, D. (2024, April 6). Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead. ArXiv.org. https://doi.org/10.48550/arXiv.2404.02525