CENTER FOR CYBER DEFENSE AND INFORMATION SECURITY

ASSERT

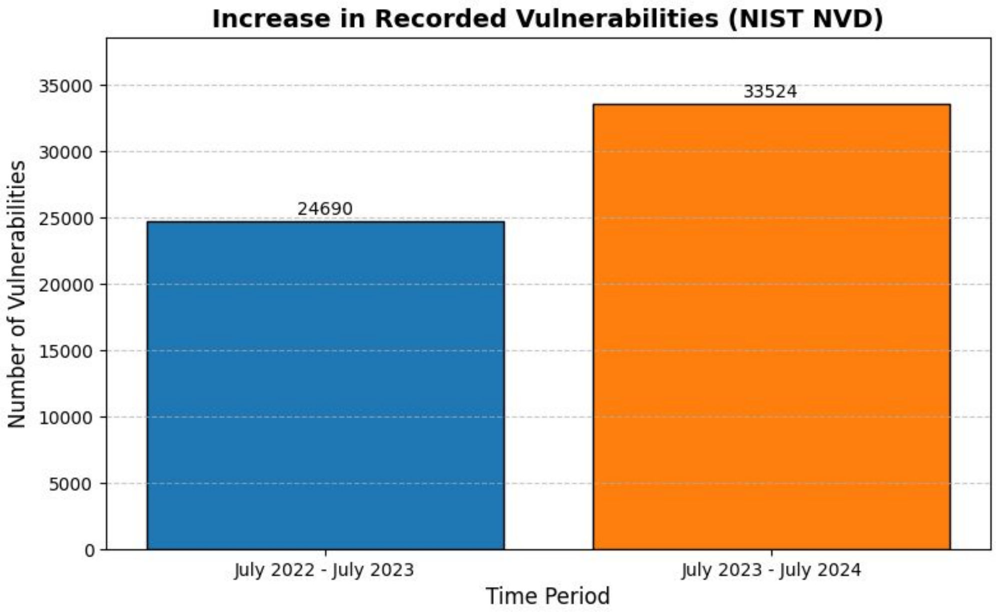
Software Vulnerability Detection with Machine Learning

Vivi Andersson supervised by Prof. Martin Monperrus, Musard Balliu

7 January 2024

Introduction

Software Vulnerabilities are Prevalent



Recorded CVEs as reported by ENISA

[1] European Union Agency for Cybersecurity (ENISA). (2024). ENISA Threat Landscape 2024.

Two Perspectives of Exploitability

Known Vulnerabilities

America's Cyber Defense Agency
NATIONAL COORDINATOR FOR CRITICAL INFRASTRUCTURE SECURITY AND RESILIENCE

CISA Known Exploited Vulnerabilities (KEV) Catalog [3]

- Vulnerabilities are inevitable but..
- Even patched ones persist in projects [2]
- Malicious actors exploiting vulnerabilities within two years after public disclosure [2]
  - The 2021 Log4Shell (ACE) vulnerability top 15 exploited in 2023 [2]

CYBERPERSONS | CYBERPANEL

CVE-2024-51378

CyberPanel Incorrect Default Permissions Vulnerability: *CyberPanel contains an incorrect default permissions vulnerability that allows for authentication bypass and the execution of arbitrary commands using shell metacharacters in the statusfile property.*

Related CWE: CWE-276

Known To Be Used in Ransomware Campaigns? **Known**

Economic costs

[2] Cybersecurity and Infrastructure Security Agency (CISA) et al., "2023 Top Routinely Exploited Vulnerabilities," Cybersecurity Advisory AA24-317A, Nov., 2024.
[3] Cybersecurity and Infrastructure Security Agency (CISA), "Known Exploited Vulnerabilities Catalog," Online.

Zero-day Vulnerabilities

- Undisclosed vulnerabilities (unknown)

- CVE-2022-42475: Heap-based **Buffer Overflow** [4]
  - FortiOS SSL-VPN component
  - Remote Code Execution without authentication

[2] Cybersecurity and Infrastructure Security Agency (CISA) et al., “2023 Top Routinely Exploited Vulnerabilities,” Cybersecurity Advisory AA24-317A, Nov., 2024.
[4] Fortinet PSIRT, “Importance of Patching: An Analysis of the Exploitation of N-Day Vulnerabilities,” Fortinet Blog, Online.

KTH VETENSKAP OCH KONST

```
$"A"x100000' > payload
$curl --data-binary @payload -H 'Content-Length: 32+1 bits
https://vpn.example.com:8443/remote/logincheck?AAAA=BBBB'
```

ter is a
(overflo

Malicious request to open endpoint → process input payload

```
char* sslvpn_ap_pcalloc(pool* myPool, int64_t requestedSize){
...
uint64_t alignedSize = (((requestedSize - 1) / 8 ) + 1) * 8;
if ( &info->nextFreeSpace[alignedSize] > endOfAllocation ) {
  // There is not enough space left. We must allocate a new chunk.
else {
  // Assume sufficient space in chunk. Allocate.
  result = myPool->info->nextFreeSpace;
  myPool->info->nextFreeSpace += alignedSize;
  }
}
```

Requires unsigned integer, leading to **integer underflow** → `alignedSize` is big

Conditional evaluates to false, indicating enough space in pool.

Now points outside actual pool (payload was larger!)

```
Program received signal SIGSEGV, Segmentation fault.
Payload stored at RAX (return address)
```

Find way redirect program flow

Zero-day Vulnerabilities

Data and Credential Stealing

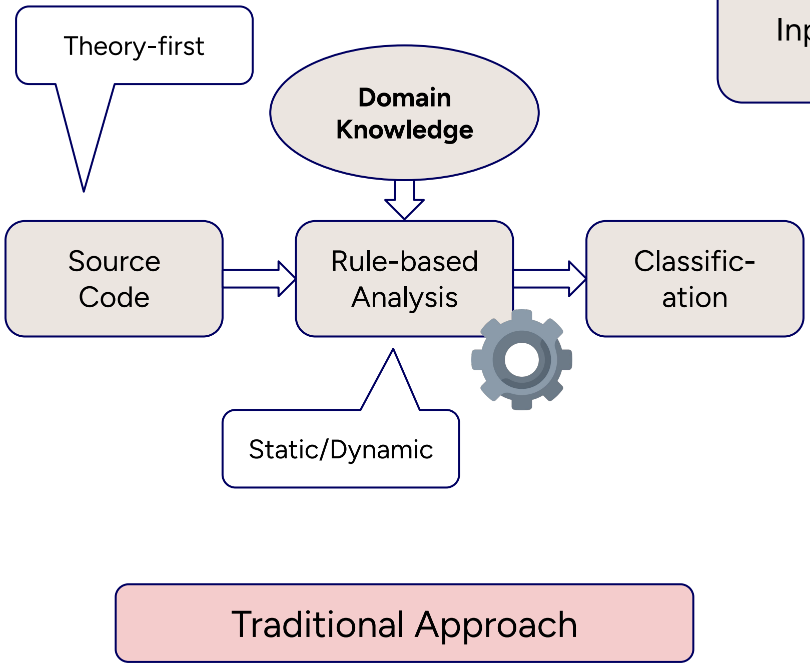
- Undisclosed vulnerabilities (unknown)
- CVE-2022-42475: Heap-based **Buffer Overflow** [4]
  - FortiOS SSL-VPN component
  - Remote Code Execution without authentication
  - Exploited by APT actors targeting governmental and strategic targets
- The majority of exploited vulnerabilities In 2023 were zero-day [2]

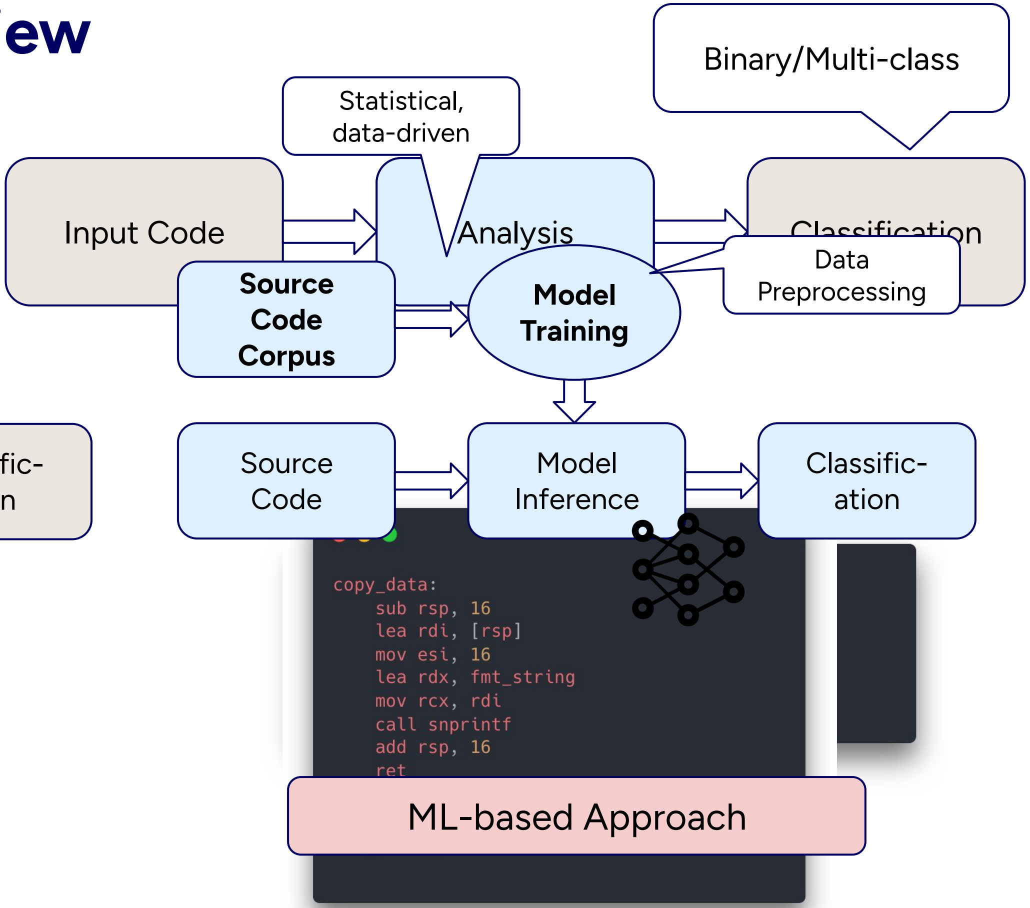
[2] Cybersecurity and Infrastructure Security Agency (CISA) et al., “2023 Top Routinely Exploited Vulnerabilities,” Cybersecurity Advisory AA24-317A, Nov., 2024.
[4] Fortinet PSIRT, “Importance of Patching: An Analysis of the Exploitation of N-Day Vulnerabilities,” Fortinet Blog, Online.

Software vulnerabilities are pervasive and costly, making their detection crucial.

Automated Software Vulnerability Detection (SVD)

SVD: Task and Overview





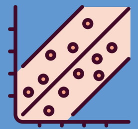
ML-Based Software Vulnerability Detection

Scoping "Machine Learning"

Machine Learning

Learns **how** to perform specific tasks for structured data analysis and prediction

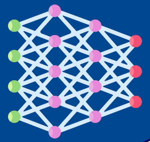
E.g. linear regression



Deep Learning

Learns hierarchical **features** for complex pattern recognition

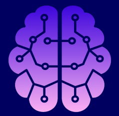
E.g. graph neural networks



Foundation Models

Learns **advanced functionalities** for task-agnostic problem solving

E.g. GPT-4, BERT, …



Hierarchy inspired by [5]

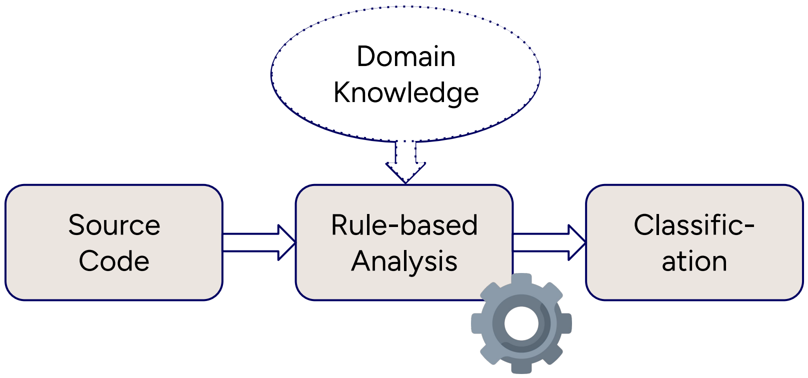
[5] R. Bommasani et al., "On the opportunities and risks of foundation models," arXiv:2108.07258 cs.LG, Aug. 2021.

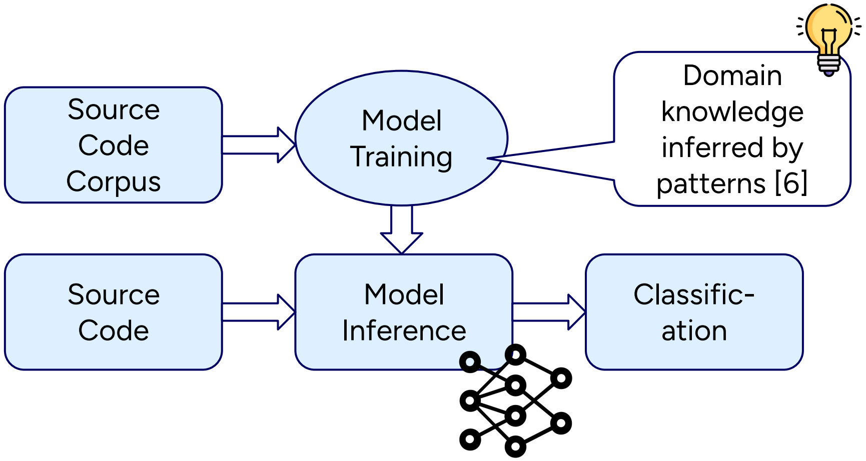
Some Promises of ML for SVD

Data-Driven Knowledge & Adaptability

- Challenge: Domain knowledge
  - Problem may be too complex to model symbolically [6]
- Challenge: Reliance on predefined rules
  - E.g. new vulnerabilities

Foundation models show generalisability **to out-of-distribution patterns** [5] with little (e.g. transfer learning [7]) or no retraining (eg. in-context learning [8]).



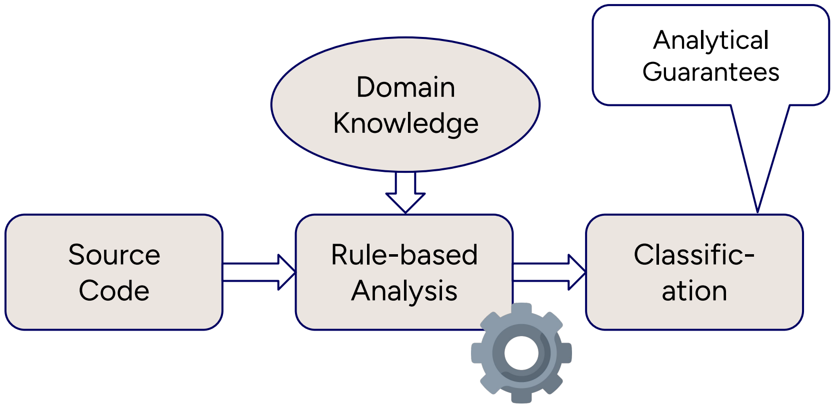


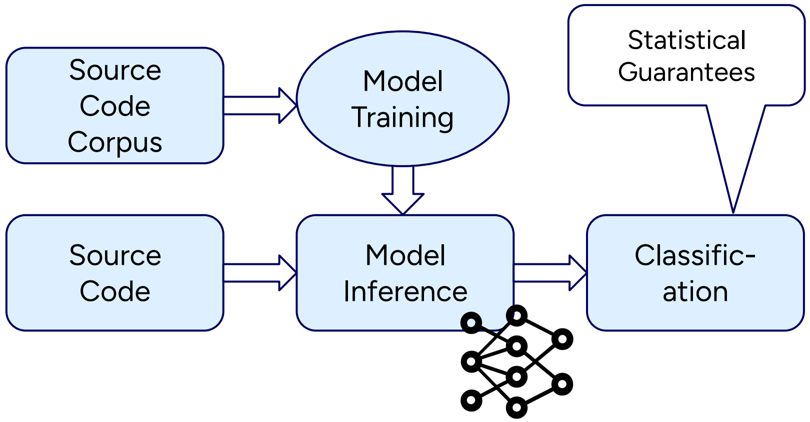
[5] R. Bommasani et al., “On the opportunities and risks of foundation models,” arXiv:2108.07258 cs.LG, Aug. 2021. [6] O. Simeone, Machine Learning for Engineers. Cambridge, 2022.
[7] Z. Chen, S. Kommrusch, and M. Monperrus, “Neural transfer learning for repairing security vulnerabilities in c code,” IEEE Trans. Softw. Eng. (2022)
[8] M. Geng et al., “Large Language Models are Few-Shot Summarizers: Multi-Intent Comment Generation via In-Context Learning,” IEEE Trans. Softw. Eng. (2024).

Scalability

- Challenge: Analysis at scale [9 ¾]
  - Static analysis: false positives
  - Dynamic analysis: false negatives

Probabilistic reasoning provides a good **average case** that is scalable [9].



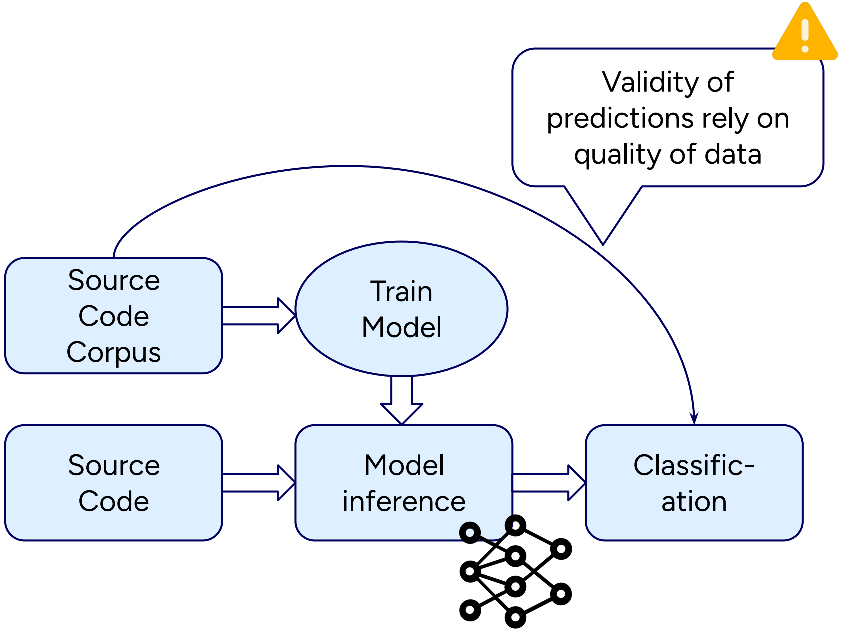


[9] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, "A Survey of Machine Learning for Big Code and Naturalness," ACM Comput. Surv., vol. 51, 2018.
[9 ¾] J. Smith, B. Johnson, E. Murphy-Hill, B. Chu, and H. R. Lipford, "Questions developers ask while diagnosing potential security vulnerabilities with static analysis," in Joint Meeting on Foundations of Software Engineering. ACM (2015)

Current State & Challenges

Data Challenges

- Data availability
  - Need labelled data samples
  - Scarce for safety-critical industries [8]
- Data representativeness [11, 12]
  - Label inconsistencies are prevalent
  - Synthetic data
  - Short code snippets
- Strong results on benchmark datasets
  - Degradation on realistic data [10-12]

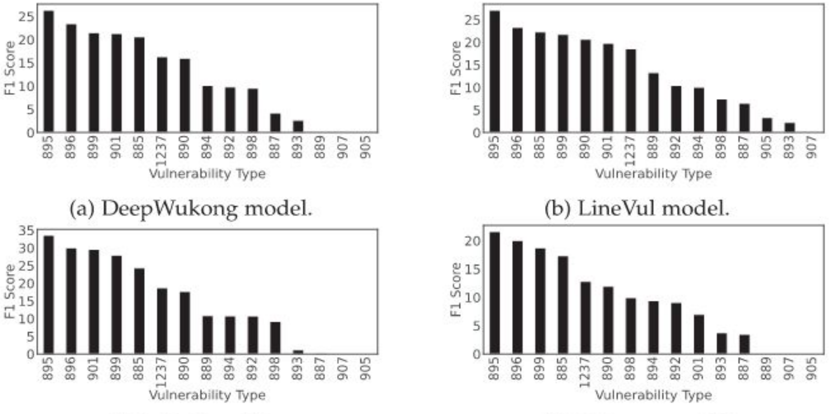


[10] Y. Chen, Z. Ding, L. Alowain, X. Chen, and D. Wagner, "DiverseVul: A New Vulnerable Source Code Dataset for Deep Learning Based Vulnerability Detection," in RAID '23. ACM. (2023)
[11] Y. Ding et al., "Vulnerability Detection with Code Language Models: How Far Are We?," Jul. 10, 2024, arXiv: arXiv:2403.18624
[12] P. Chakraborty, K. K. Arumugam, M. Alfadel, M. Nagappan, and S. McIntosh, "Revisiting the Performance of Deep Learning-Based Vulnerability Detection on Realistic Datasets," *IIEEE Trans. Software Eng.*, 2024.

Recent Data Advancements

- More realistic datasets (mainly C) [10-12]
  - Diverse vulnerab: 63 [12] & 93 [11] CWEs
- Realistic evaluation
  - **GNNs** (a, c-d) [11]
  - **Transformer-based** (b) [11]
  - LLMs [10]
- Performance varies by vulnerability type [12]
  - Strong: Information leaks, tainted input
  - Weaker: Complex/contextual patterns
    1. CWE-416: Use after free
    2. CWE-893: Path traversal (site.com/input:=.../etc/pwd)
    3. CWE-343: Predictable value ranges
- LLMs need more work [11] and specialized methods (transformer) currently outperforms them [12]

Performance per Software Fault Pattern (SFPs group similar CWEs) [12]



(a) DeepWukong model. (b) LineVul model.

(c) ReVeal model. (d) IVDetect model.

Recent advancements provide foundation for better evaluation of vulnerability detection with machine learning

[10] Y. Chen, Z. Ding, L. Alowain, X. Chen, and D. Wagner, "DiverseVul: A New Vulnerable Source Code Dataset for Deep Learning Based Vulnerability Detection," in RAID '23. ACM. (2023)

[11] Y. Ding et al., "Vulnerability Detection with Code Language Models: How Far Are We?," Jul. 10, 2024, arXiv: arXiv:2403.18624

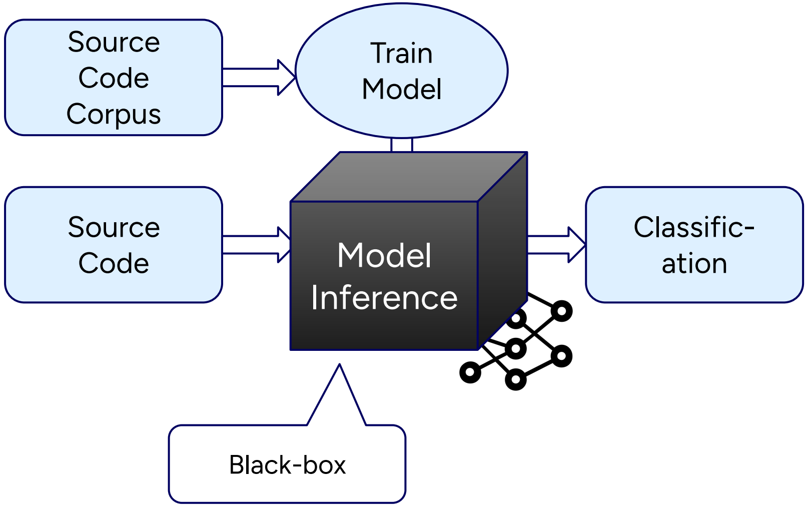
[12] P. Chakraborty, K. K. Arumugam, M. Alfadel, M. Nagappan, and S. McIntosh, "Revisiting the Performance of Deep Learning-Based Vulnerability Detection on Realistic Datasets," *IIEEE Trans. Software Eng.*, 2024.

Result Interpretability

R. Munroe, “Machine Learning,” xkcd.com, comic #1838, 2017.

- Lack of explainability
  - ML-architectures act as black boxes [6]
  - Predictions without insight into why or how the decision was made

- Impact on Security Community
  - Acceptance barrier [13]
  - Compliance, Governments etc.



[13] D. Bhusal et al., “Sok: Modeling explainability in security analytics for interpretability, trustworthiness, and usability,” in Proc. 18th Int. Conf. Availability, Reliability and Security, (2023).

Software vulnerabilities are pervasive and costly, making their detection crucial.

ML for SVD show promise but face challenges like data quality, overfitting and interpretability.

Future Research

Machine Interpretability

- Machine Interpretability
  - Breaking down **internal** reasoning of black box models
- XAI
  - Human-understandable explanations of model decisions
  - FOI 2019 [14]

- Why are they black box models?
  - Neural Polysemanticity
    - Neurons having multiple meanings

FOI

Explainable Artificial Intelligence: Exploring XAI Techniques in Military Deep Learning Applications

LINUS J. LUOTSINEN, DANIEL OSKARSSON, PETER SVENMARCK, ULRIKA WICKENBERG BOLIN

[14] L. J. Luotsinen, D. Oskarsson, P. Svenmarck, and U. Wickenberg Bolin, “Explainable Artificial Intelligence: Exploring XAI Techniques in Military Deep Learning Applications,” Swedish Defence Research Agency (FOI), Stockholm, Sweden, Tech. Rep. FOI-R–4849–SE, Dec. 2019.

State of the art machine interpretability: **Sparse Autoencoders** [16]

(a) abstraction (lexical and perceptual)

(d) polysemanticity

Neurons firing on **conceptually related** but distinct features [15]

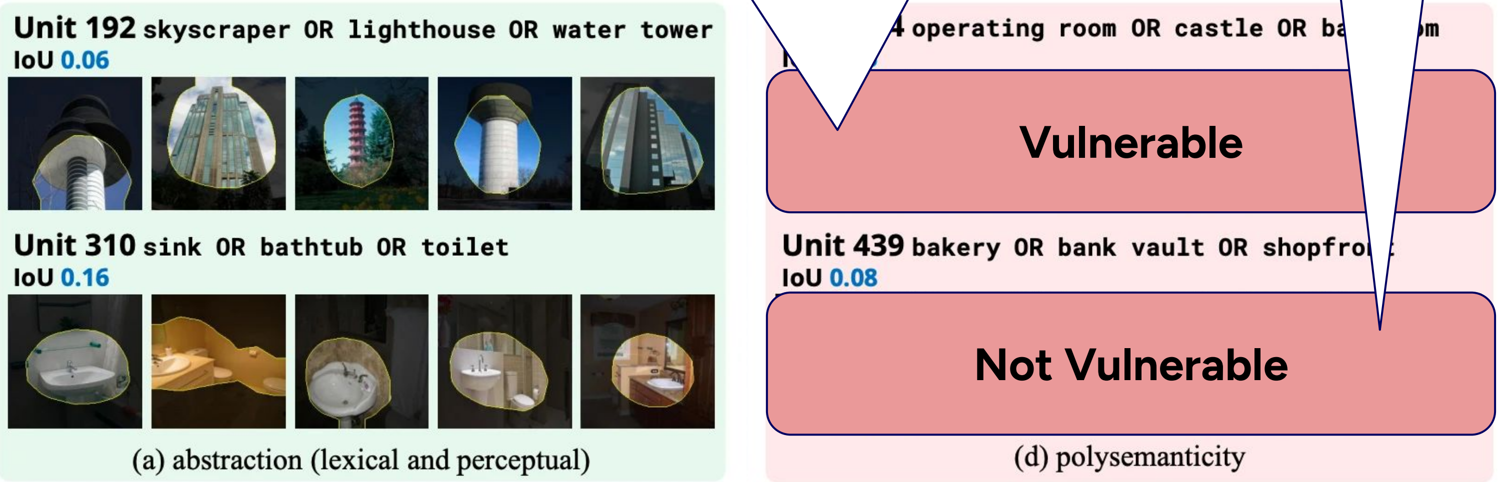
Neurons firing on multiple, **unrelated** features [15]

[15] Mu, Jesse, Jacob Andreas. "Compositional explanations of neurons." *Advances in Neural Information Processing Systems* 33 (2020)
[1] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, “Sparse Autoencoders Find Highly Interpretable Features in Language Models,” Oct. 04, 2023, arXiv

For Vulnerability Detection…

5 rows of code **AND** buffer length 16 **OR** banana

buffer length > 36



Vulnerable

Not Vulnerable

(a) abstraction (lexical and perceptual)

(d) polysemanticity

Decomposition of model’s reasoning in vulnerability classification can guide better designs

High accuracy in vulnerability detection can be achieved even when only word counts are available [17]

[15] Mu, Jesse, Jacob Andreas. "Compositional explanations of neurons." *Advances in Neural Information Processing Systems* 33 (2020)
[17] N. Risse and M. Böhme, “Top Score on the Wrong Exam: On Benchmarking in Machine Learning for Vulnerability Detection,” Aug. 23, 2024.

Hybrid Symbolic/Neural Approaches

- Program Analysis as preprocessing input for ML-model
  - Multimodality of ML inputs: source code + diagnostic
- Machine Learning to guide Program analysis
  - Guiding Fuzzing (DARPA AI Cyber Challenge [17])
  - Formally verified C Code with LLM (ASSERT/SCANIA) [18]
- Program Analysis as a Verification of LLMs output

Integrating ML (LLMs) with traditional program analysis for more rigorous code analysis

Machine Learning or Traditional Approach?

[18] Defense Advanced Research Projects Agency (DARPA), "AI Cyber Challenge," DARPA.mil. Online.
[19] M. Sevenhuijsen, K. Etemadi, and M. Nyberg, "VeCoGen: Automating Generation of Formally Verified C Code with Large Language Models," arXiv preprint, 2024

Software vulnerabilities are pervasive and costly, making their detection crucial.

ML for SVD show promise but face challenges like data quality, overfitting and interpretability.

Future research should focus on explaining machine learning predictions and improving detection capabilities.

Cryptographic API Misuse in Go

Current Project

Project Overview

- Problem
  - Cry
  - Imp
    -
    -
    -
- Focus on
  - Misı [20]
  - Go used in security critical domains such as Kubernetes, coreDNS
  - **Recent case:** CVE-2024-45337 SSH authentication bypass due to "widely-misused" Go API

```go
type MyEncrypter struct {
    KeyLength int
}
// simple misuse
var BasicLength = 1024

func GenKey(e *MyEncrypter) *rsa.PrivateKey {
    privateKey, _ := rsa.GenerateKey(rand.Reader, e.KeyLength+BasicLength)
    return privateKey
}
```

: Failures”
10 most
bilities [17]

a misuse

[17] OWASP Foundation, “A02:2021 – Cryptographic Failures,” OWASP Top 10:2021. Online.
[18] A.-K. Wickert, L. Baumgärtner, M. Schlichtig, K. Narasimhan, and M. Mezini, “To Fix or Not to Fix: A Critical Study of Crypto-misuses in the Wild,” (TrustCom). IEEE, (2022)
[19] W. Li, S. Jia, L. Liu, F. Zheng, Y. Ma, and J. Lin, “CryptoGo: Automatic Detection of Go Cryptographic API Misuses,” in Computer Security Applications Conf. ACM. (2022)
[20] Y. Zhang, et al. “Gopher: High-Precision and Deep-Dive Detection of Cryptographic API Misuse in the Go Ecosystem,” ACM CCS’24 (2024)

Related

- Improve documentation
- Automatic detection of misuses [18-20]
    - Mainly Java/Android, rule-based
- **Recent Go work:** Gopher [20]
    - Detects 19 different misuses
    - Static Data Flow Analysis (def-use)

[ref] Y. Zhang, et al. “Gopher: High-Precision and Deep-Dive Detection of Cryptographic API Misuse in the Go Ecosystem,” ACM CCS’24 (2024)

Research Gap (WIP)

- False alarms
    - Non-security critical context
    - Non-exploitable
- Over approximation (soundness)
    - Less practical reports?

Can we leverage ML learning abilities to detect more complex misuse patterns?

Conclusion

Software vulnerabilities are pervasive and costly, making their detection crucial.

ML for SVD show promise but face challenges like data quality, overfitting and interpretability.

Future research should focus on explaining machine learning predictions and improving detection capabilities.