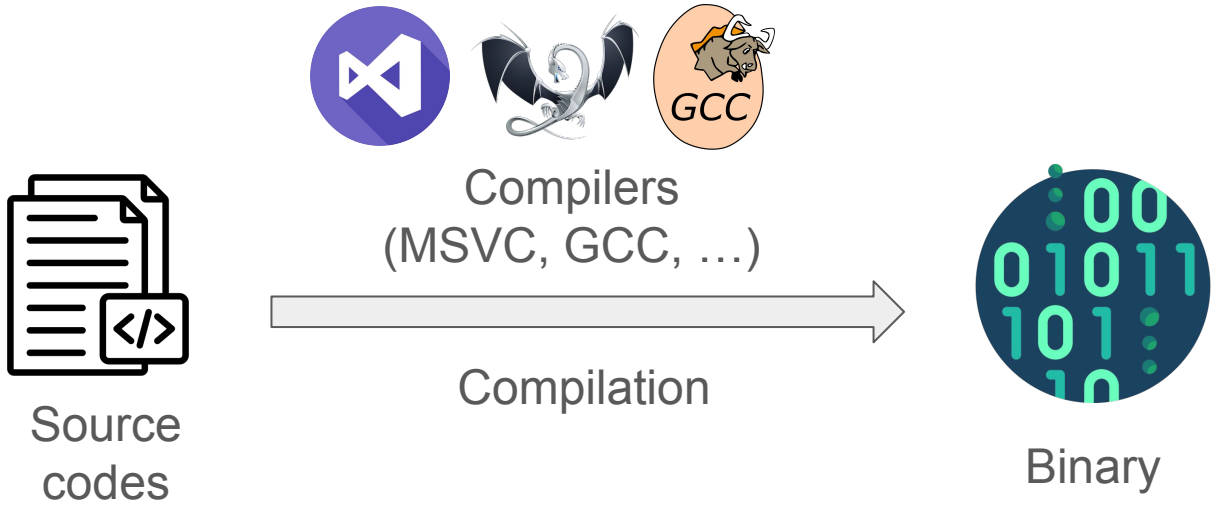
NEURAL INFORMATION PROCESSING SYSTEMS

ASSEMBLAGE: Automatic Binary Dataset Construction for Machine Learning

Chang Liu*, Rebecca Saul*, Yihao Sun , Edward Raff, Maya Fuchs, Townsend Southard Pantano, James Holt, Kristopher Micinski

Syracuse University
UMBC
LPS LABORATORY FOR PHYSICAL SCIENCES
Booz | Allen | Hamilton®

What is binary

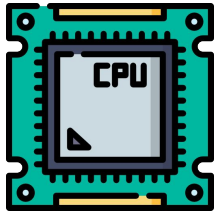


Why binary analysis

Source codes

Human readable

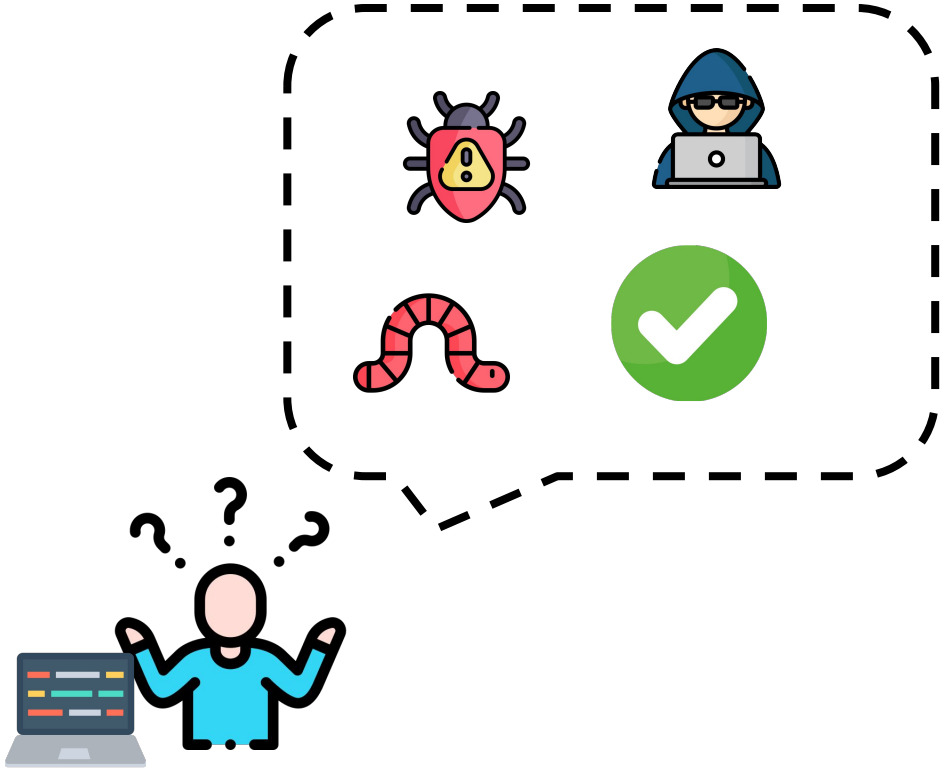
Binary file



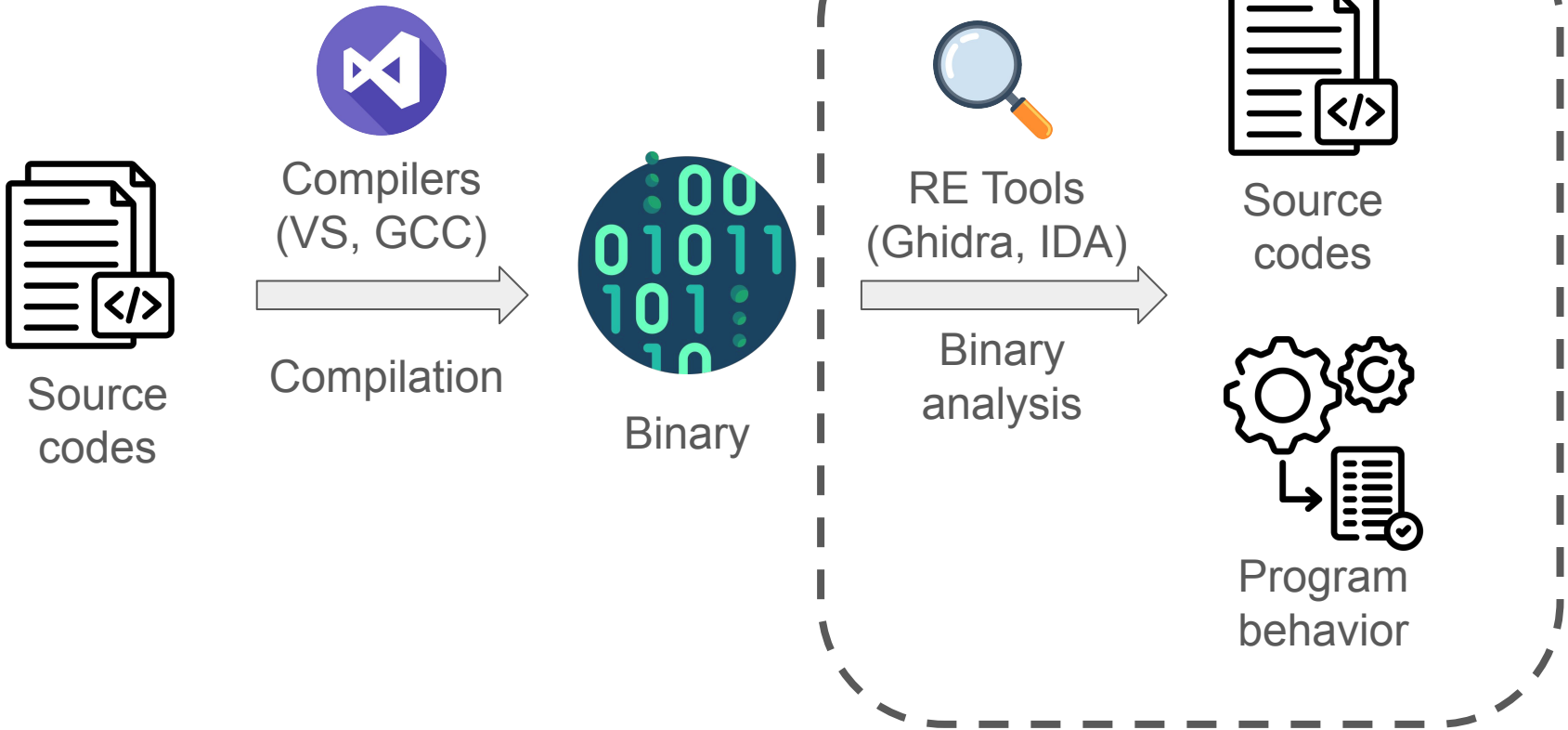
Not human readable, only machine readable

Why binary analysis

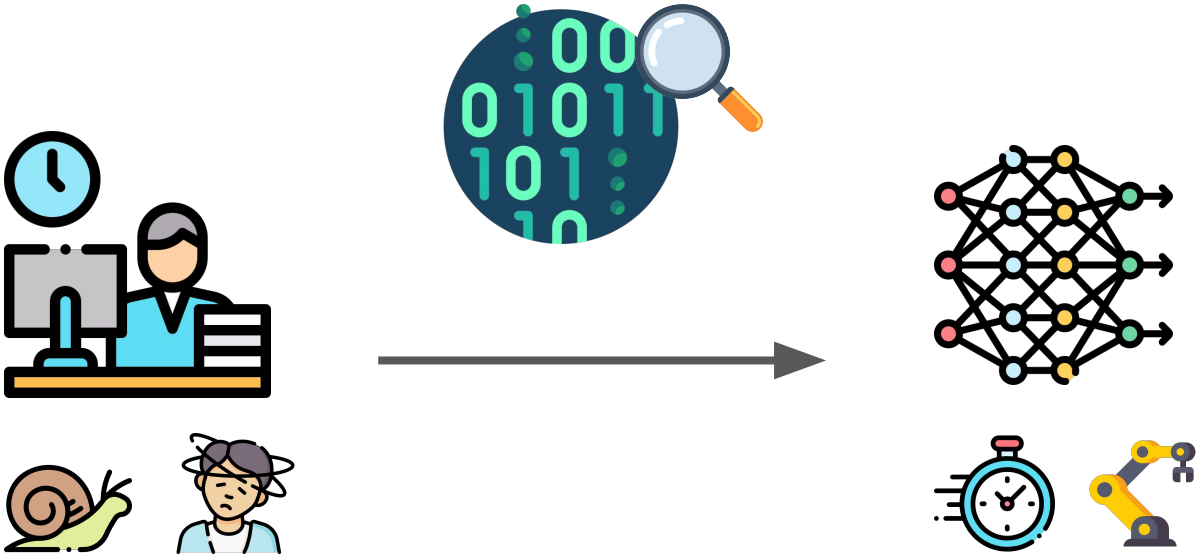
Not human readable
Runs everywhere
Running at all time



What is binary analysis



Why machine learning based binary analysis



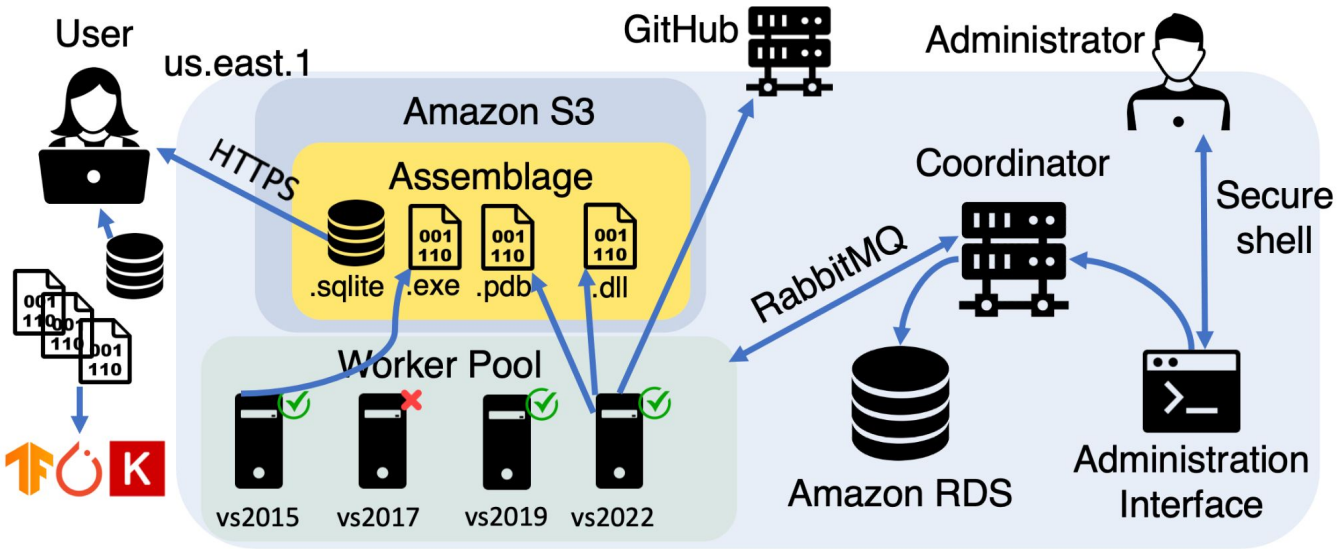
Slow, labor intensive

Fast, automated, scaleable

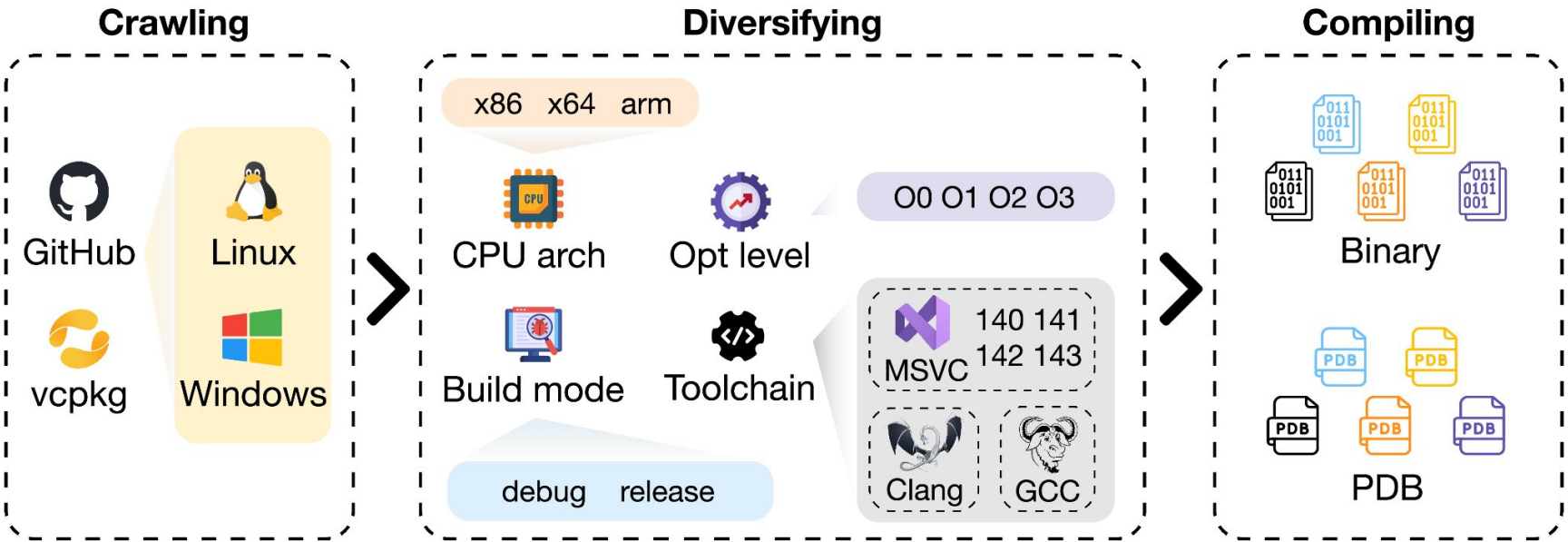
Where is the dataset?

Dataset	Binaries (#)	Functions (#, k)	Projects (#)	OS	Compilers
SPEC CPU	981	?	7		3
Ubuntu dataset	87,853	88,000	22,040		2
BinKit	243,128	75,231	51		2
BinaryCorp26M	48,130	25,877	9,819		2
BinBench	1,127,479	4,408	?	 	2
Assemblage	**1,536,171**	**783,694**	**220,792**		**3**

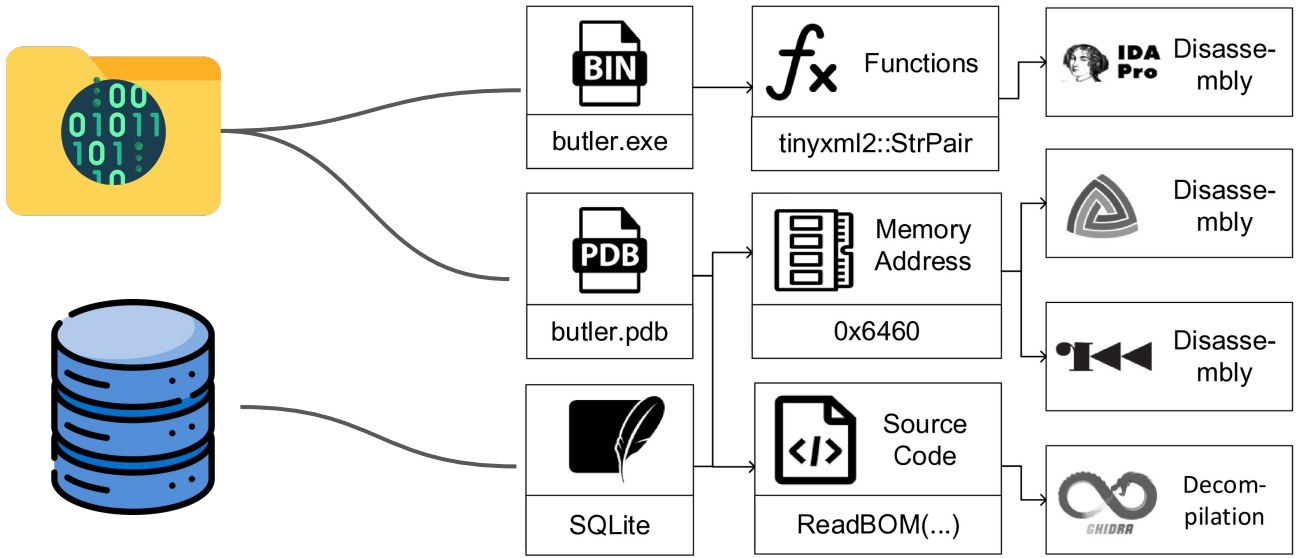
Assemblage - architecture



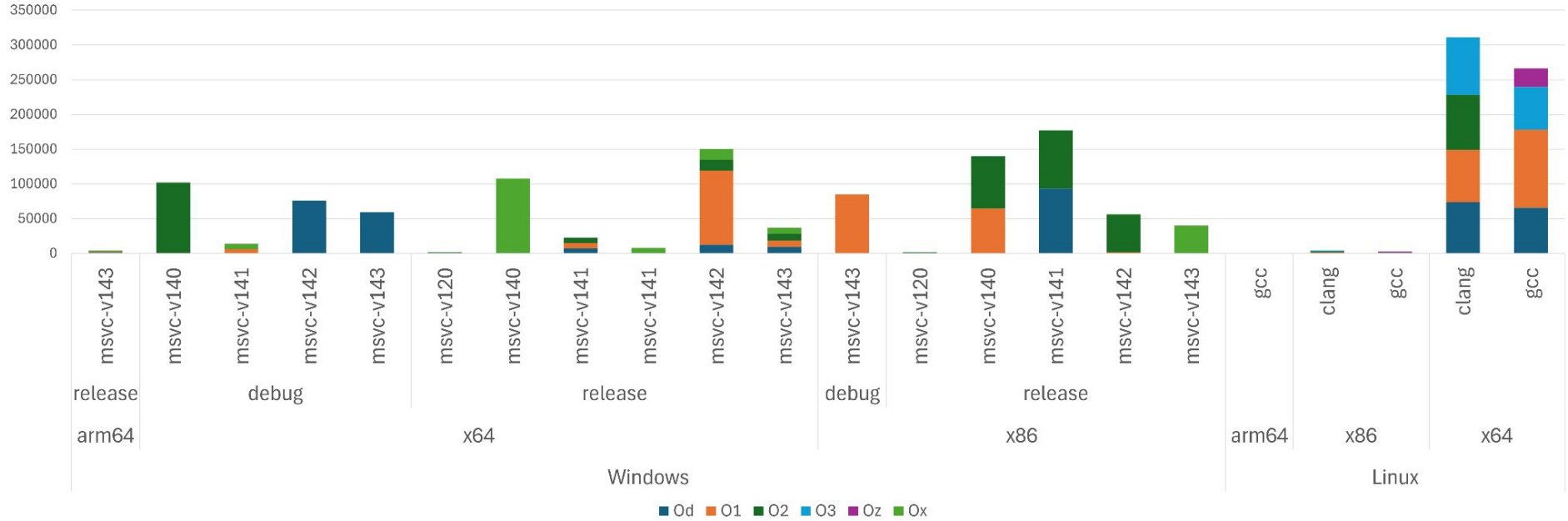
Assemblage - binary build pipeline



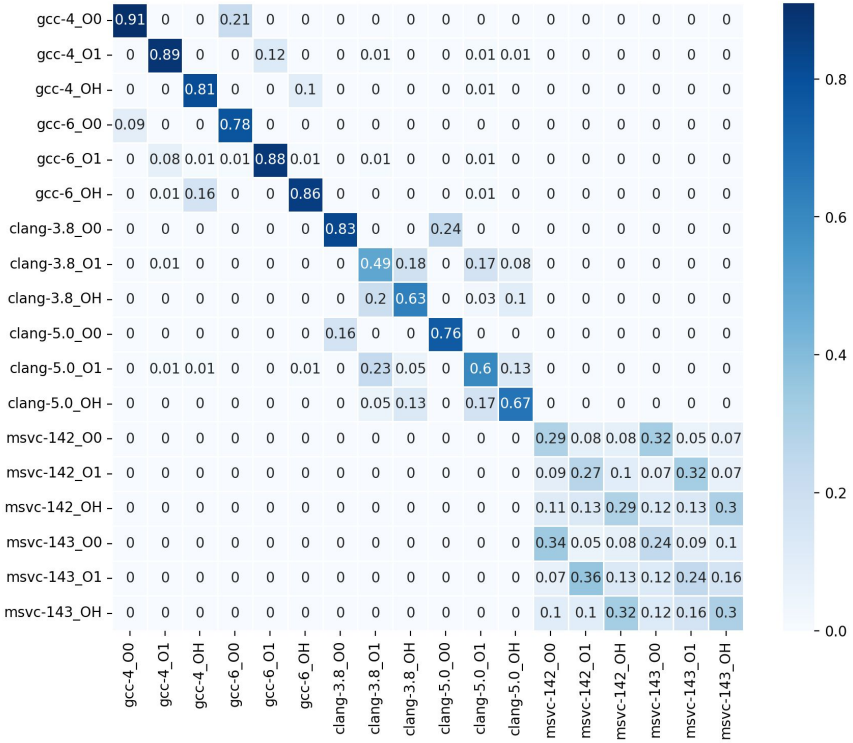
Assemblage - dataset overview



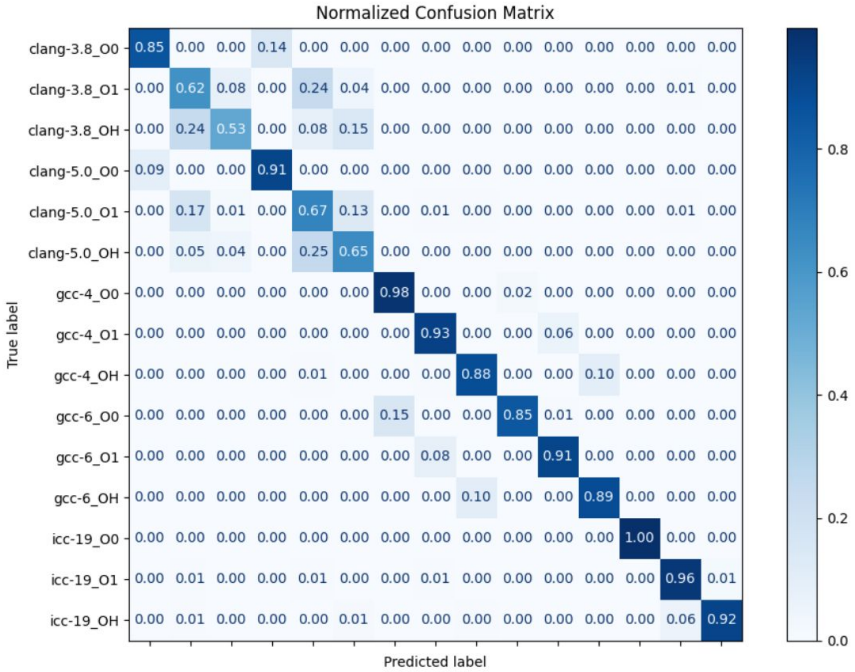
Assemblage - dataset stats



Benchmark: compiler provenance with PassTell



Train/test on Assemblage data



Train/test on PassTell data

Benchmark: function similarity with GNN

Task	Win-Linux	Linux-Win	Linux-Linux	Win-Win
arch	0.50	-	0.97	-
bit	0.70	-	0.99	-
comp	0.63	-	0.80	-
opt	0.72	0.62	0.88	0.89
ver	0.82	0.64	0.98	0.84
XA	0.48	-	0.86	-
XC	0.63	-	0.86	-
XC+XB	0.61	-	0.87	-
XM	0.54	0.61	0.87	0.86

arch: CPU architecture
bit: 32-bit or 64 bit
comp: compiler
opt: optimization
ver: compiler version
XA: only different architectures and bitness
XC: only same architecture and bitness
XC+XB: only same architecture
XM: come from arbitrary architectures, bitness, compiler, compiler versions, and optimization.

Benchmark: function similarity with jTrans

Training epochs	1	20	50
Fine-tune & Evaluate on Windows	0.17	0.52	0.52
Fine-tune & Evaluate on Linux	0.83	0.83	0.83
Fine-tune & Evaluate on BinCorp-26M	0.82	-	0.98

Benchmark: function boundary identification with XDA

Dataset	XDA	IDA	IDA (w/ pdb)
Assemblage Windows (GitHub)	0.75	0.47	0.79
Assemblage Windows (vcpkg)	0.81	0.86	0.86

Public access to Assemblage (codes and dataset)

https://github.com/Assemblage-Dataset/Assemblage

https://assemblagedocs.readthedocs.io/en/latest/dataset.html#id1

References (partial, full references in paper)

A. Marcelli, M. Graziano, X. Ugarte-Pedrero, Y. Fratantonio, M. Mansouri, and D. Balzarotti. How machine learning is solving the binary function similarity problem. In 31st USENIX Security Symposium (USENIX Security 22), pages 2099–2116, Boston, MA, Aug. 2022. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/marcelli.

Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli. Graph matching networks for learning the similarity of graph structured objects. In K. Chaudhuri and R. Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 3835–3845. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/li19d.html.

H. Wang, W. Qu, G. Katz, W. Zhu, Z. Gao, H. Qiu, J. Zhuge, and C. Zhang. jtrans: jump-aware transformer for binary code similarity detection. Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, 2022. URL https://api.semanticscholar.org/CorpusID:249062999.

Y. Du, K. Snow, F. Monrose, et al. Automatic recovery of fine-grained compiler artifacts at the binary level. In 2022 USENIX Annual Technical Conference (USENIX ATC 22), pages 853–868, 2022.

K. Pei, J. Guan, D. W. King, J. Yang, and S. Jana. Xda: Accurate, robust disassembly with transfer learning. In Proceedings of the 2021 Network and Distributed System Security Symposium (NDSS), 2021.

Thank you