Shaky structures: The wobbly world of causal graphs in software analytics

Hallunications? Worse than you think

Jeremy Hulse (NCSU)
Nasir U. Eisty (U.Tennesse)
Tim Menzies (NCSU)

prof, cs, **ncstate**, usa
http://timm.fyi

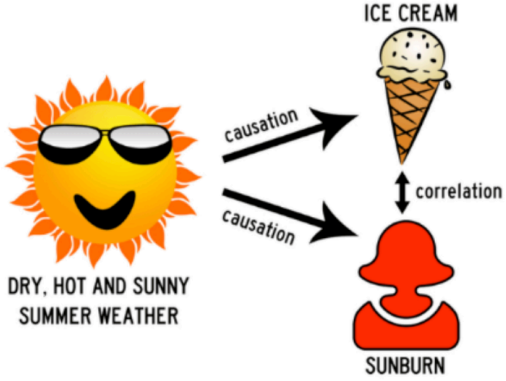
EMSE, 30(142), 2025, https://doi.org/10.1007/s10664-025-10690-6
SLIDES: http://timm.fyi/26cause.pdf

Apr 16 2026

Claim: Causal graph generations not a source of ”truth”

- **We don’t want** to know that ”Size correlates with Defects.”
- We want to know: ”If I change this module, will I stop the bugs?”
- Causal Discovery Algorithms (CDAs) promise to find these hidden structures automatically.
- But... suffer from **mass hallunication.**

Ronald Fisher:
Correlation is not Causation
(but can we access "causation"?)



Cigaret Cancer Link Is Bunk

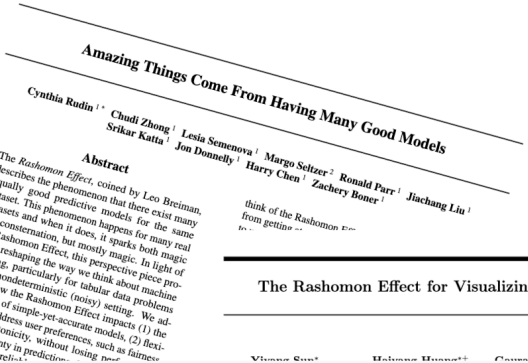
By CHARLES GOLDEN

70 Million Smokers Have Been Victims Of a False Alarm

Rashomon Volumes

Given so many good models, epistemically dishonest to reason about "the" model

(a) Rashomon set of the 1,365 best sparse decision trees for t
COMPAS dataset, generated by TreeFARMS an
TimberTrek (figure from Wang et al. 20



Amazing Things Come From Having Many Good Models

Cynthia Rudin [1] · Chudi Zhong [1] · Lesia Semenova [1] · Margo Seltzer [2] · Ronald Parr [1] · Jiachang Liu [1]
Srikar Katta [1] · Jon Donnelly [1] · Harry Chen [1] · Zachery Boner [1]

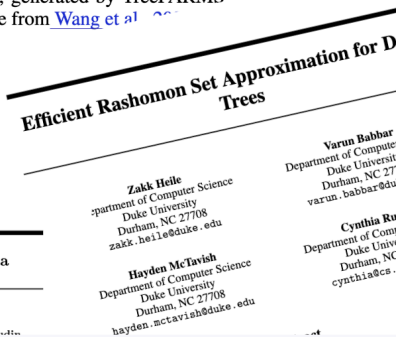
Abstract

The *Rashomon Effect*, coined by Leo Breiman, describes the phenomenon that there exist many equally good predictive models for the same dataset. This phenomenon happens for many real datasets and when it does, it sparks both magic and consternation, but mostly magic. In light of the Rashomon Effect, this perspective piece proposes reshaping the way we think about machine learning, particularly for tabular data problems in the nondeterministic (noisy) setting. We address how the Rashomon Effect impacts (1) the existence of simple-yet-accurate models, (2) flexibility to address user preferences, such as fairness and monotonicity, without losing pre
(3) uncertainty in prediction
nations, (4) relia
gorith

think of the Rashomon Ef
from getting

The Rashomon Effect for Visualizing High-Dimensional Data

Yiyang Sun[1] Haiyang Huang[1*] Cynthia Rudin[1]



Efficient Rashomon Set Approximation for D
Trees

Zakk Heile
Department of Computer Science
Duke University
Durham, NC 27708
zakk.heile@duke.edu

Hayden McTavish
Department of Computer Science
Duke University
Durham, NC 27708
hayden.mctavish@duke.edu

Varun Babbar
Department of Computer
Duke Universit
Durham, NC 27
varun.babbar@du

Cynthia Ru
Department of Comp
Duke Univ
Durham, NC
cynthia@cs.

The Reality: The Instability Problem

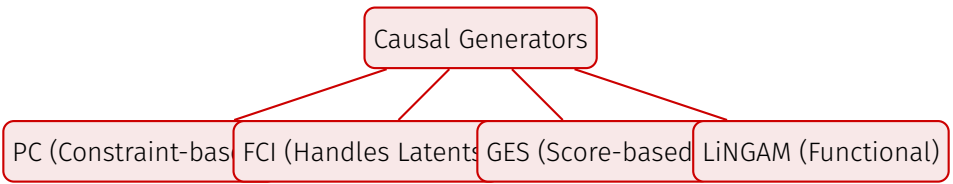
Claim: Causal graphs generators: **too unstable** to trust.

The Definition of Instability

Small changes to the data result in **large, unpredictable changes** in the resulting graphs.

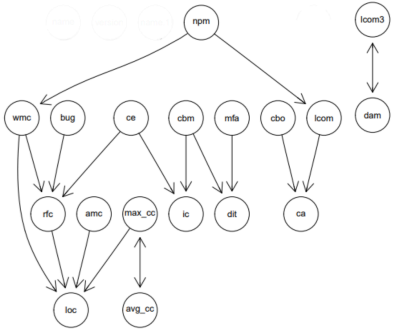
- If the graph changes every time you re-run it, it isn't an "explanation."
- It is just **noise masquerading** as a diagram.

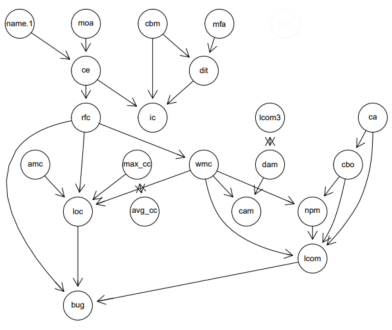
Four most prominent paradigms in causal discovery: all hallunicate



- $J(A,B) = \frac{|E_A \cap E_B|}{|E_A \cup E_B|}$
- $J = 1.0$: The graphs are **identical**.
- $J = 0.0$: The graphs have zero edges in common.
- **Rule:** If J is low, your "Why Engine" is broken. [cite: 25]

We measured Jaccard across versions, across tools, and across data subsets. e.g. v5,v7:





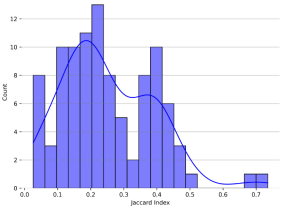
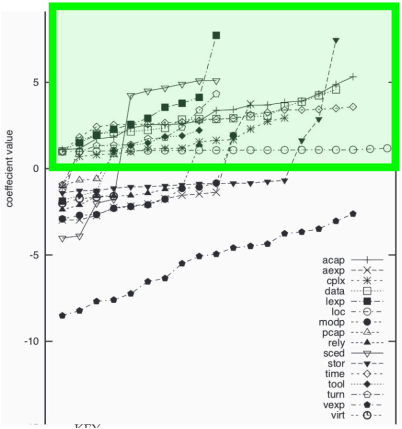


Fig. 2: Jaccard index results seen in 100 experiments where two causal graph generators are applied to the same data set.

Hallunications in SE (regression, effort estimation)

20 times,

- **Select** 66% of the data;
- Learn model $effort = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$
- Show the **sorted** β_i values.



Hallunications in SE (defect prediction)

In 28 publications:

- Is feature X postively or negatively **correalted** to defects?

ref	cbo	rfc	lcom	dit	noc	wmc	#projects	size	type
[13]	+	+	+	-	-	+	6	95-201 classes	6 versions of rhino (java)
[14]	+	+	+	-	-	+	12	86 classess (3-12kloc)	student
[15]	+	+	-				1	1700 classes (110kloc)	commercial telecom
[16]	+	+	-	+	+	+	8	113 classes	student
[17]	+	+	-	+	+	+	8	114 classes	student
[18]	+	+	+	+	-		1	83 classes	commercial: lalo (c++)
[19]				+	+		1	32 classes	commercial: telecom c++
[20]				+	-		1	42-69 classes	commercial java word proc.
[21]	+	-	-	-	-	-	1	85 classes	telecom c++
[22]	-	+		-	-	+	3	92 classes	3 c++ subsystems,commercial
[23]	+	+	+	-	-	+	1	123 classes (34kloc)	java commercial
[24]	+			+		+	1	706 classes	commercial c++ and java
[25]	+	+	+	-	+	+	1	145 classes	kc1-nasa
[26]	+	+	+	+	-	+	1	3677 classes	open source:mozilla
[27]	+	+	+			+	1	?	java (sap) commercial
[28]	+	+	+	+	+	+	3	?	eclipse 2.0, 2.1, 3.0
[29]	-	+	+	-	-	+	8	113 classes	student
[30]		+	+	+	+		2	64 classes	sales and cd-selection system
[31]		-		-	-	-	1	3344 modules (2mloc)	commercial telecom c++
[32]	+	+	+	-	-	+	5	395 classes	commercial telecom c++
[33]	+	+		-	-	+	1	1412 classes	open source:jdt
[34]	+	+		-	-	+	2	9713 classes	eclipse 2.0, 2.1
[35]	+	+	-	-	-	+	1	145 classes	kc1-nasa
[36]				+	-		1	145 classes	commercial java xml editor
[37]	-	-	-	-	-	-	1	174 classes	commercial telecom c++
[38]	-					-	0	50 classes	student
[39]	+	+	-	-	-	+	1	145 classes	kc1-nasa
[40]		+		+	+		2	294 classes	commercial c++
total +	18	20	11	11	8	17			
total -	4	3	7	14	16	4			

Total percents: "*" denotes majority conclusion in each column

+	* 64%	* 71%	* 39%	39%	29%	* 61%
-	14%	11%	25%	* 50%	* 57%	14%

KEY:

- Strong consensus (over 2/3rds)
- Some consensus (less than 2/3rds)
- Weak consensus (about half)
- No consensus

Hallunications in SE (topic modeling)

Groups of words (topics) seen in Stackoverflow:

- Two runs, **shuffle** row order in-between;
- Column1 = percent of words in run1's topics appearing in **run2's topics**

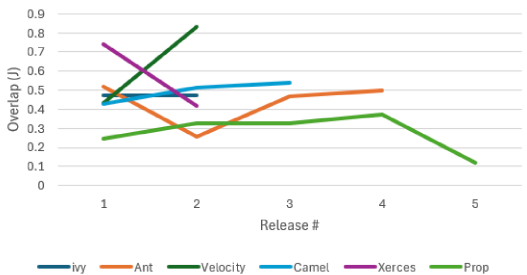
%	Topic	Top Words
100	Xaml Binding	grid window bind...
100	Ruby on Rails	rubi rail gem lib...
100	MVC	http com java...
88	Objective C	self cell nsstring...
88	Func. Returns	function const char...
77	Testing	chang tri work use...
77	Socket Comm.	Main send socket...
77	Q and A	know need want...
77	OO Prog.	instanc void type...
77	HTML Links	com http www url...

%	Topic	Top Words
77	Display	height jpg imag...
66	Windows Tips	studio net dll web...
66	Website Des.	left span style text...
66	Web Dev.	script jqueri func...
66	Java Prog.	java void new...
66	Date/Time	select day valu...
66	Database	column order group...
66	Android View	height content text...
66	.NET Frame.	net form text check...
55	iOS App Dev.	applic user need...

Finding 1: The Time Flip in defect prediction data

Question: Does the ”cause” of defects stay the same from Version 1 to Version 2?

- **The Result:** In nearly all cases, over half the edges vanish between releases.
- Median stability across versions: **Extremely Low**.
- **The Risk:** A manager fixes a ”cause” in Version 1.5 that no longer exists in Version 1.7.



Finding 2: The Domain Gap

Question: If I learn the causal graph for Camel, can I use it for Tomcat?

- **The Result:** Over two-thirds of causal connections differ between projects.
- This suggests causal discovery in SE is **highly idiosyncratic**.
- Generalization is a dangerous assumption in this landscape.

Table 6: **RQ2 results:** Inter-Project Jaccard Values. This matrix is symmetrical across the diagonal (so the values of blank cell i, j can be found in cell j, i).

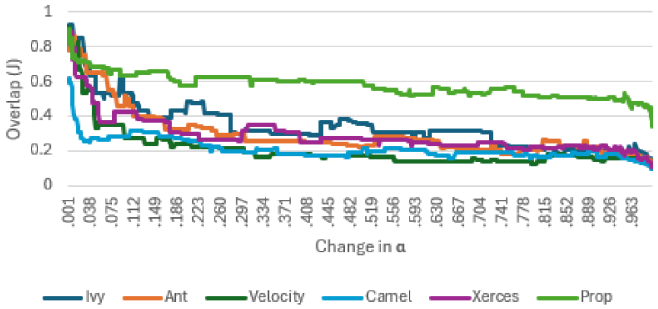
Project	Ivy	Ant	Velocity	Camel	Xerces
Ivy	1	.28	.37	.25	.26
Ant		1	.28	.19	.20
Velocity			1	.20	.26
Camel				1	.29
Xerces					1

Finding 3: The Fragility of Alpha

Question: Does graph change with changing significance level α (e.g., 0.01 to 0.05)?

- **The Result:** Minor tuning adjustments generate radically different causal structures.
- These tools are **not robust** to standard hyperparameter variations.

Fig. 4: **RQ3 results:** Graph of Jaccard values when shifting alpha $\Delta = 0.001$. All graphs compared to the graph from $\alpha = 0.001$.



Finding 5: Other Data? Other Causal Generators?

$$\underbrace{\begin{pmatrix} RQ1 \\ RQ2 \\ RQ3 \\ RQ4 \end{pmatrix}}_{4 \text{ research questions}} \times \underbrace{\begin{pmatrix} PC \\ FCI \\ GES \\ LiNGAM \end{pmatrix}}_{4 \text{ generators}} \times \underbrace{\begin{pmatrix} 18 \text{ defect prediction} \\ 2 \text{ software configuration} \\ 3 \text{ software process management} \end{pmatrix}}_{3 \text{ tasks, } 23 \text{ data sets}}$$

Table 7: Jaccard indexes seen in different experiments.

		Causal graph generator			
data	**question**	**PC** μ (σ)	**FCI** μ (σ)	**GES** μ (σ)	**LiNGAM** μ (σ)
defect	**RQ1** : across releases	.67 (.07)	.43 (.11)	.50 (.08)	.39 (.06)
defect	**RQ2** : across projects	.42 (.04)	.19 (.07)	.34 (.04)	.24 (0.5)
defect	**RQ3** : changing α	.51 (.06)	.27 (.07)	N/A	N/A
config	**RQ3** : changing α	.27 (.07)	.10 (.04)	N/A	N/A
process	**RQ3** : changing α	.12 (.03)	.06 (.02)	N/A	N/A
defect	**RQ4** : subsets	.84 (.06)	.59 (.12)	.70 (.07)	.52 (.10)
config	**RQ4** : subsets	.75 (.18)	.59 (.26)	.73 (.16)	.57 (.20)
process	**RQ4** : subsets	.64 (.11)	.54 (.13)	.59 (.09)	.29 (.07)

The Why: Why Do Graphs Fail in SE?

- **Data Sparsity:** SE data is often ”fat and short” (many features, few rows). [cite: 9]
- **Noisy Metrics:** Static code features are proxies, not direct physical laws.
- **Stochastic Nature:** Software evolution is driven by human choice, not just mathematical necessity. [cite: 37]

We need **hybrid workflows** that combine these graphs with symbolic checks. [cite: 5, 25]

Roadmap: Moving Beyond the ”Hype”

1. **Intersection Filters:** Only report edges found by *at least two* different generators.
2. **Stability Audits:** Always run a ”10% drop test” before publishing a graph.
3. **Type-Safe Checks:** Use Z3 to reject causal links that violate known SE constraints. [cite: 14, 25]
4. **Ensemble Causality:** Treat causal discovery as an optimization problem over an ensemble of graphs. [cite: 33]

Conclusion

- Causal graphs are a powerful **hypothesis generator**, but a dangerous **decision maker**—until we solve the instability problem.

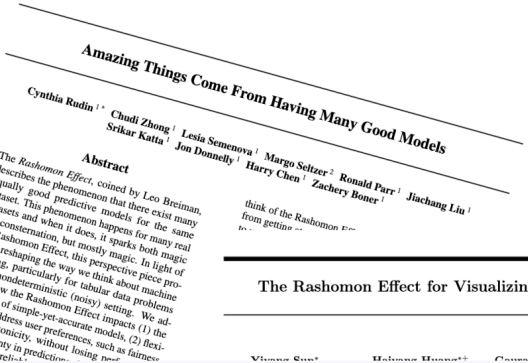
Closing pitch

The hallunication problem is much **worse** than you think.

Rashomon Volumes

Given so many good models, epistemically dishonest to reason about "the" model

(a) Rashomon set of the 1,365 best sparse decision trees for t
COMPAS dataset, generated by TreeFARMS an
TimberTrek (figure from Wang et al. 20



Amazing Things Come From Having Many Good Models

Cynthia Rudin [1] · Chudi Zhong [1] · Lesia Semenova [1] · Margo Seltzer [2] · Ronald Parr [1] · Jiachang Liu [1]
Srikar Katta [1] · Jon Donnelly [1] · Harry Chen [1] · Zachery Boner [1]

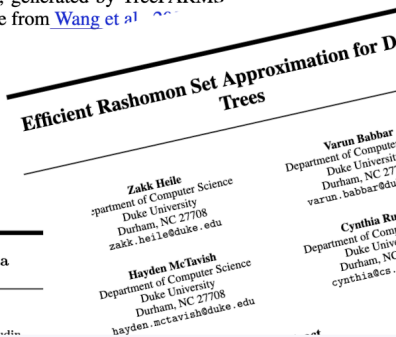
Abstract

The *Rashomon Effect*, coined by Leo Breiman, describes the phenomenon that there exist many equally good predictive models for the same dataset. This phenomenon happens for many real datasets and when it does, it sparks both magic and consternation, but mostly magic. In light of the Rashomon Effect, this perspective piece proposes reshaping the way we think about machine learning, particularly for tabular data problems in the nondeterministic (noisy) setting. We address how the Rashomon Effect impacts (1) the existence of simple-yet-accurate models, (2) flexibility to address user preferences, such as fairness and monotonicity, without losing pre
(3) uncertainty in prediction
nations, (4) relia
gorith

think of the Rashomon Ef
from getting

The Rashomon Effect for Visualizing High-Dimensional Data

Yiyang Sun[1] Haiyang Huang[1*] Cynthia Rudin[1]



Efficient Rashomon Set Approximation for D
Trees

Zakk Heile
Department of Computer Science
Duke University
Durham, NC 27708
zakk.heile@duke.edu

Hayden McTavish
Department of Computer Science
Duke University
Durham, NC 27708
hayden.mctavish@duke.edu

Varun Babbar
Department of Computer
Duke Universit
Durham, NC 27
varun.babbar@du

Cynthia Ru
Department of Comp
Duke Univ
Durham, NC
cynthia@cs.