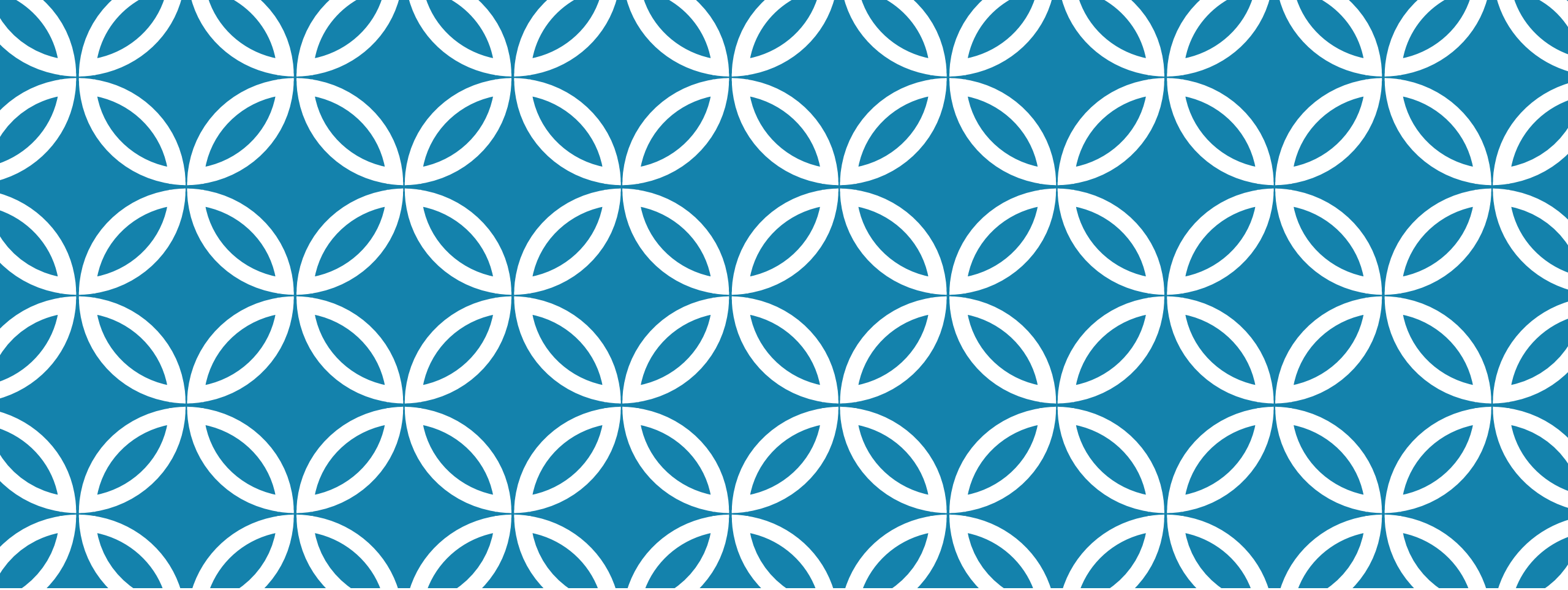


REVERSE ENGINEERING 17 CARS IN UNDER 10 MINUTES

BRENT STONE

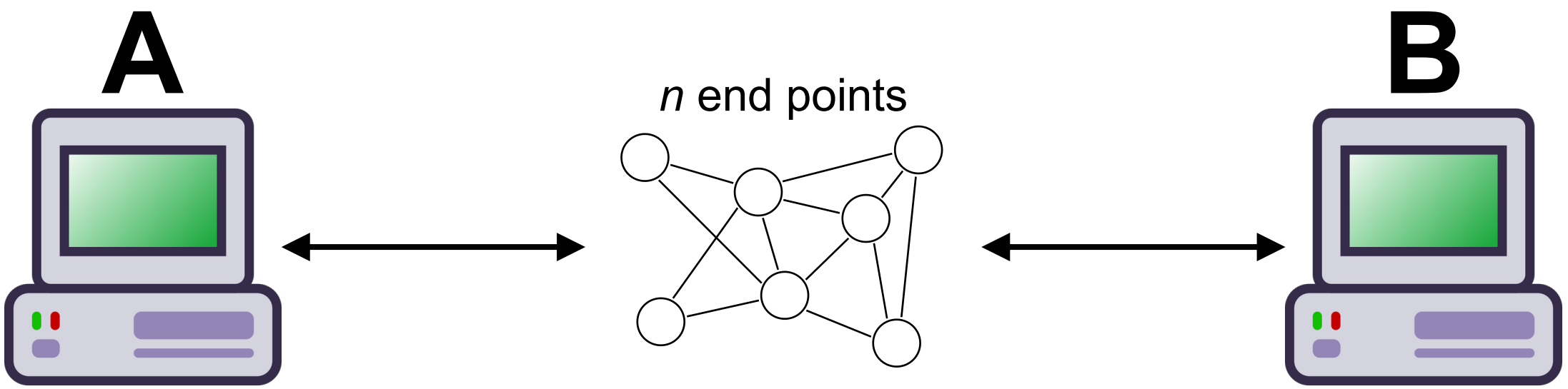
Disclaimer About This Talk and The Github Repo

The views expressed in this presentation are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Army, the United States Department of Defense or the United States Government. The material publicly released on https://github.com/brent-stone/CAN_Reverse_Engineering/, up to and including commit ac0e55f on 26 March 2019, is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Case Numbers: 88ABW-2019-0910, 88ABW-2019-0024

General Use Networks



FLEXIBLE

- Modify End Points
- Modify Routing

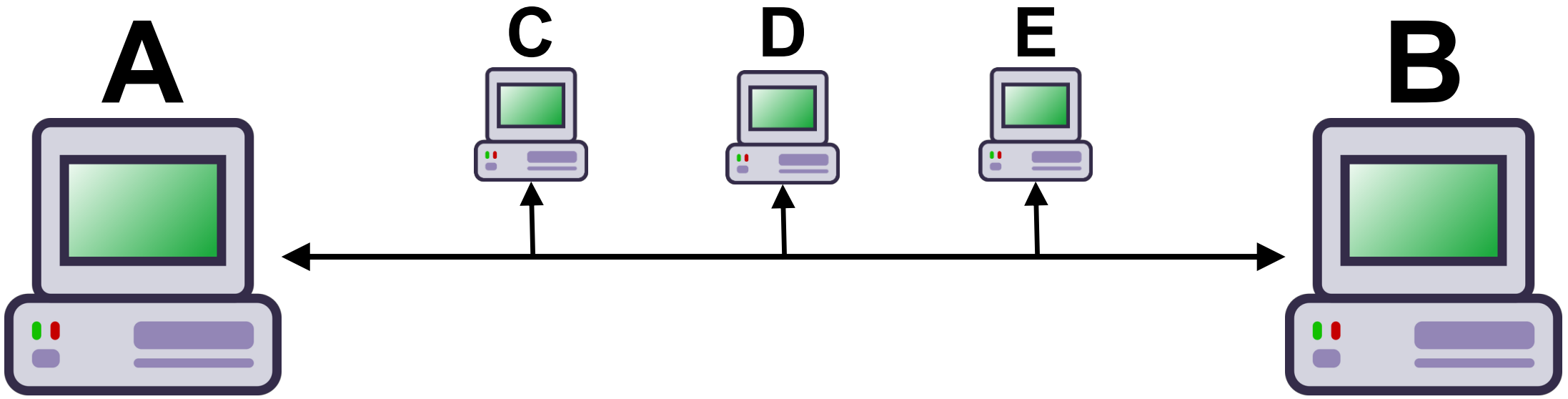
Meta Data

UNDETERMINABLE

- No delivery guarantee
- No timeliness guarantee

Meta Data

Control Networks



DETERMINABLE

- Delivery Guarantee
- Timeliness Guarantee

Meta Data

INFLEXIBLE

- Fixed End Points
- Fixed Routing

Meta Data

interwebs

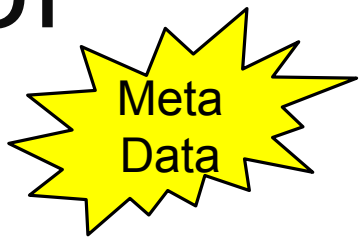
VoIP

Lots of people helping others play with general use networks…

IoT

smart phones

Automated Reverse Engineering of **General Use Networks**



1. P. Ducange, G. Mannara, F. Marcelloni, R. Pecori, and M. Vecchio, "A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classiffiers," in 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2017, pp. 1-6.

2. J. Yuan, Z. Li, and R. Yuan, "Information entropy based clustering method for unsupervised internet traffic classification," in IEEE International Conference on Communications (ICC), 2008, pp. 1588-1592.

3. C. Besiktas and H. A. Mantar, "Real-Time Traffic Classiffication Based on Cosine Similarity Using Sub-application Vectors," in Proceedings of the Traffic Monitoring and Analysis 4th International Workshop, 2012, vol. 7189, pp. 89-92.

4. A. Trifilo, S. Burschka, and E. Biersack, "Traffic to protocol reverse engineering," in IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1-8.

5. M. E. DeYoung, "Dynamic protocol reverse engineering: a grammatical inference approach," Air Force Institute of Technology, 2008.

6. W. Cui, M. Peinado, K. Chen, H. J.Wang, and L. Irun-Briz, "Tupni: Automatic Reverse Engineering of Input Formats," in 15th ACM Conference on Computer and Communications Security (CCS), 2008, pp. 391-402.

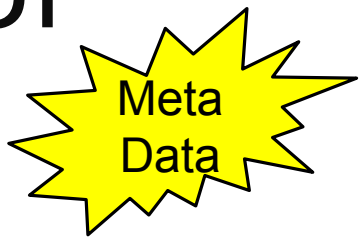
7. J. Newsome, D. Brumley, J. Franklin, and D. Song, "Replayer: automatic protocol replay by binary analysis," in 13th ACM conference on Computer and Communications Security (CCS), 2006, p. 311.

8. J. Caballero, P. Poosankam, C. Kreibich, and S. D., "Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering," in 16th ACM Conference on Computer and Communications Security (CCS), 2009, pp. 621-634.

9. J. Caballero, H. Yin, Z. Liang, and D. Song, "Polyglot: Automatic Extraction of Protocol Message Format using Dynamic Binary Analysis," in 14th ACM Conference on Computer and Communications Security (CCS), 2007, pp. 317-329.

10. W. Cui, V. Paxson, N. C. Weaver, and R. H. Katz, "Protocol-Independent Adaptive Replay of Application Dialog," in Network and Distributed System Security Symposium (NDSS), 2006, pp. 279-293.

Automated Reverse Engineering of **General Use Networks**



11. M. Wakchaure, S. Sarwade, I. Siddavatam, and P. Range, "Reconnaissance of Industrial Control System By Deep Packet Inspection," in 2nd IEEE International Conference on Engineering and Technology (ICETECH), 2016, no. 3, pp. 1093-1096.

12. J. Antunes, N. Neves, and P. Verissimo, "Reverse engineering of protocols from network traces," in 18th Working Conference on Reverse Engineering, 2011, pp. 169-178.

13. M. A Beddoe, "Network protocol analysis using bioinformatics algorithms," McAfee, Santa Clara, CA, USA, 1, 2004.

14. Y. Wang, Z. Zhang, D. Yao, B. Qu, and L. Guo, "Inferring Protocol State Machine from Network Traces: A Probabilistic Approach," in International Conference on Applied Cryptography and Network Security, 2011, pp. 1-18.

15. P. M. Comparetti, G. Wondracek, C. Kruegel, and E. Kirda, "Prospex: Protocol specification extraction," in IEEE Symposium on Security and Privacy, 2009, pp. 110-125.

16. J. Erman and M. Arlitt, "Traffic classification using clustering algorithms," in 2006 SIGCOMM Workshop on Mining Network Data, 2006, pp. 281-286.

17. F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of Eight Data Mining Algorithms for Smarter Internet of Things (IoT)," in International Workshop on Data Mining in IoT Systems (DaMIS 2016), 2016, vol. 98, no. 1, pp. 437-442.

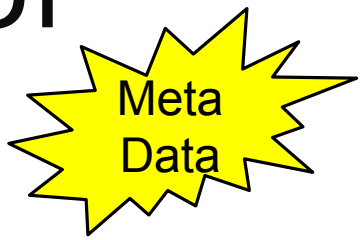
18. Y. Wang et al., "A semantics aware approach to automated reverse engineering unknown protocols," in 20th IEEE International Conference on Network Protocols (ICNP), 2012, pp. 1-10.

19. J. Roning, "PROTOS Protocol Genome Project," Oulu University Secure Programming Group, 2010. [Online]. Available: https://www.ee.oulu.fi/roles/ouspg/genome. [Accessed: 01-Jan-2017].

20. R. L. S. Puupera, "Domain Model Based Black Box Fuzzing Using Regular Languages," University of Oulu, 2010.

21. K. Choi, Y. Son, J. Noh, H. Shin, J. Choi, and Y. Kim, "Dissecting Customized Protocols: Automatic Analysis for Customized Protocols Based on IEEE 802.15.4," in 9th International Conference on Security of Information and Networks, 2016, pp. 183-193.

Automated Reverse Engineering of **General Use Networks**



22. Y. Wang, Y. Xiang, J. Zhang, and S. Yu, "A novel semi-supervised approach for network traffic clustering," in 5th International Conference on Network and System Security (NSS), 2011, pp. 169-175.

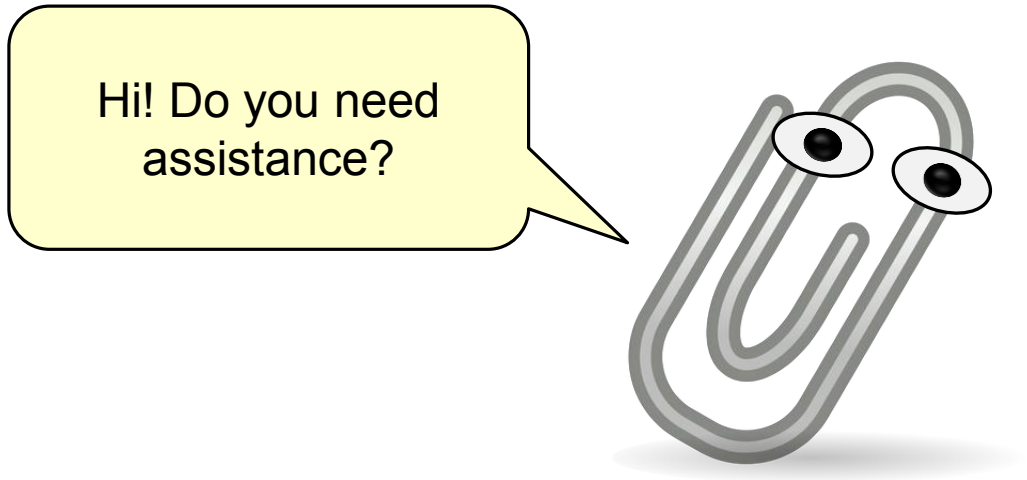
23. W. Cui, J. Kannan, and H. J. Wang, "Discoverer: Automatic Protocol Reverse Engineering from Network Traces," in USENIX Security, 2007, no. 2, pp. 199-212.

24. J. Zhang, C. Chen, Y. Xiang, and W. Zhou, "Semi-supervised and compound classiffication of network traffic," in Proceedings 32nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW), 2012, pp. 617-621.

25. T. Glennan, C. Leckie, and S. M. Erfani, "Improved Classification of Known and Unknown Network Traffic Flows Using Semi-supervised Machine Learning," in 21st Australasian Conference on Information Security and Privacy (ACISP), 2016, vol. 2, pp. 493-501.

But what about robots, cars, and other control networks?

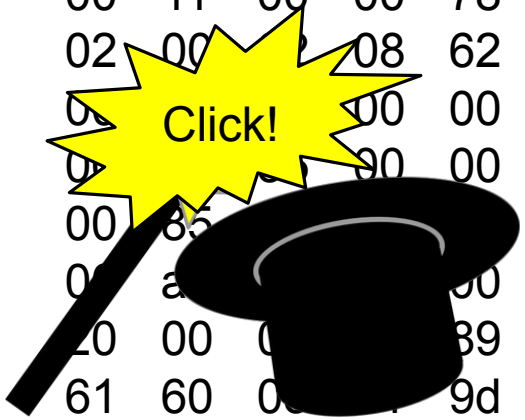
Now your computer can help!

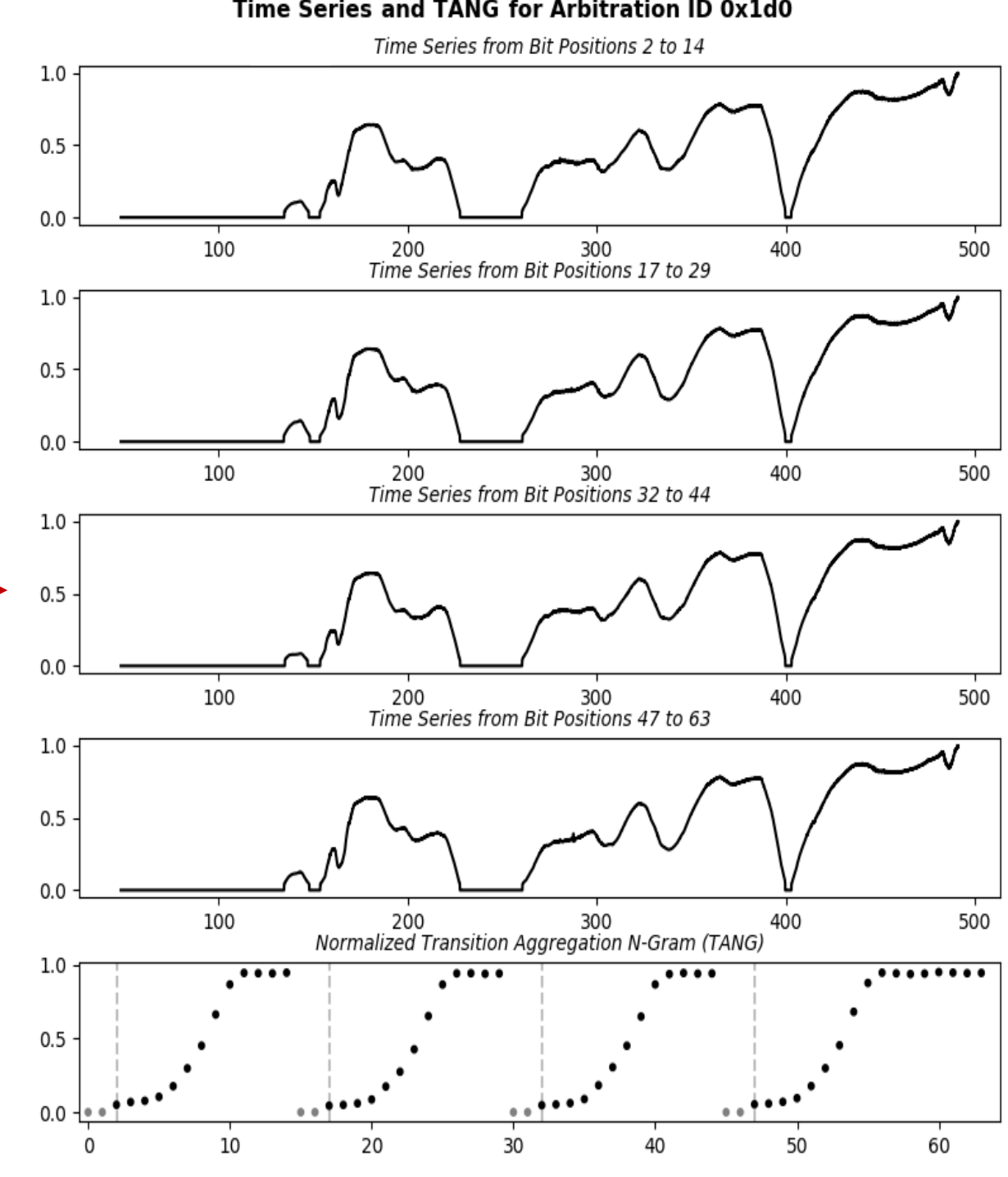


#Started canhandler on can0
#Setup complete: 48.7387
#Format:

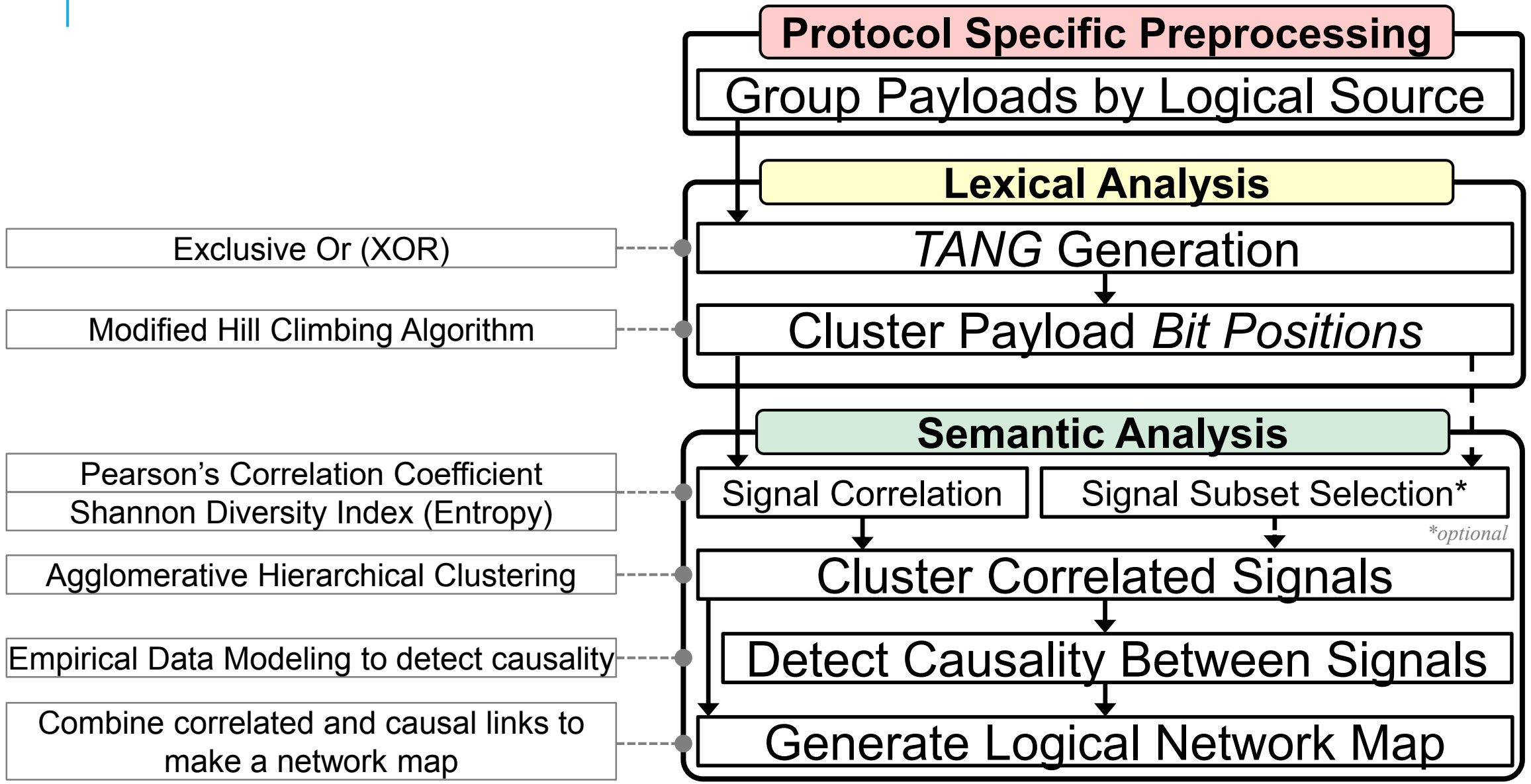
Time:	ID	DLC	Data							
48.740:	4a8	8	00	00	00	40	00	00	00	00
48.740:	020	7	00	00	07	01	00	00	2f	
48.742:	0b4	8	00	00	00	00	ac	00	00	68
48.742:	025	8	00	11	00	00	78	78	78	a6
48.743:	024	8	02	00	[illegible]	08	62	04	81	1f
48.743:	235	6	0[illegible]			00	00	3d		
48.744:	499	8	0[illegible]			00	00	00	00	00
48.745:	49a	8	00	85	[illegible]		80	28	a8	
48.746:	49b	8	0[illegible]	a[illegible]		00	00	48	10	
48.746:	262	5	20	00	0[illegible]		89			
48.747:	49d	8	61	60	0[illegible]		9d	19	c6	c5
48.747:	1c4	8	00	00	00	00	00	00	00	cd
48.749:	0aa	8	1a	6f	1a	6f	1a	6f	1a	6f
48.749:	0b6	4	00	00	00	ba				
48.749:	224	8	00	00	00	00	00	00	00	08
48.751:	127	8	68	10	00	08	00	0c	ed	a9
48.751:	020	7	00	00	07	01	00	00	2f	
48.751:	230	7	d4	43	00	00	00	00	50	
48.752:	025	8	00	11	00	00	82	82	82	c4

.....



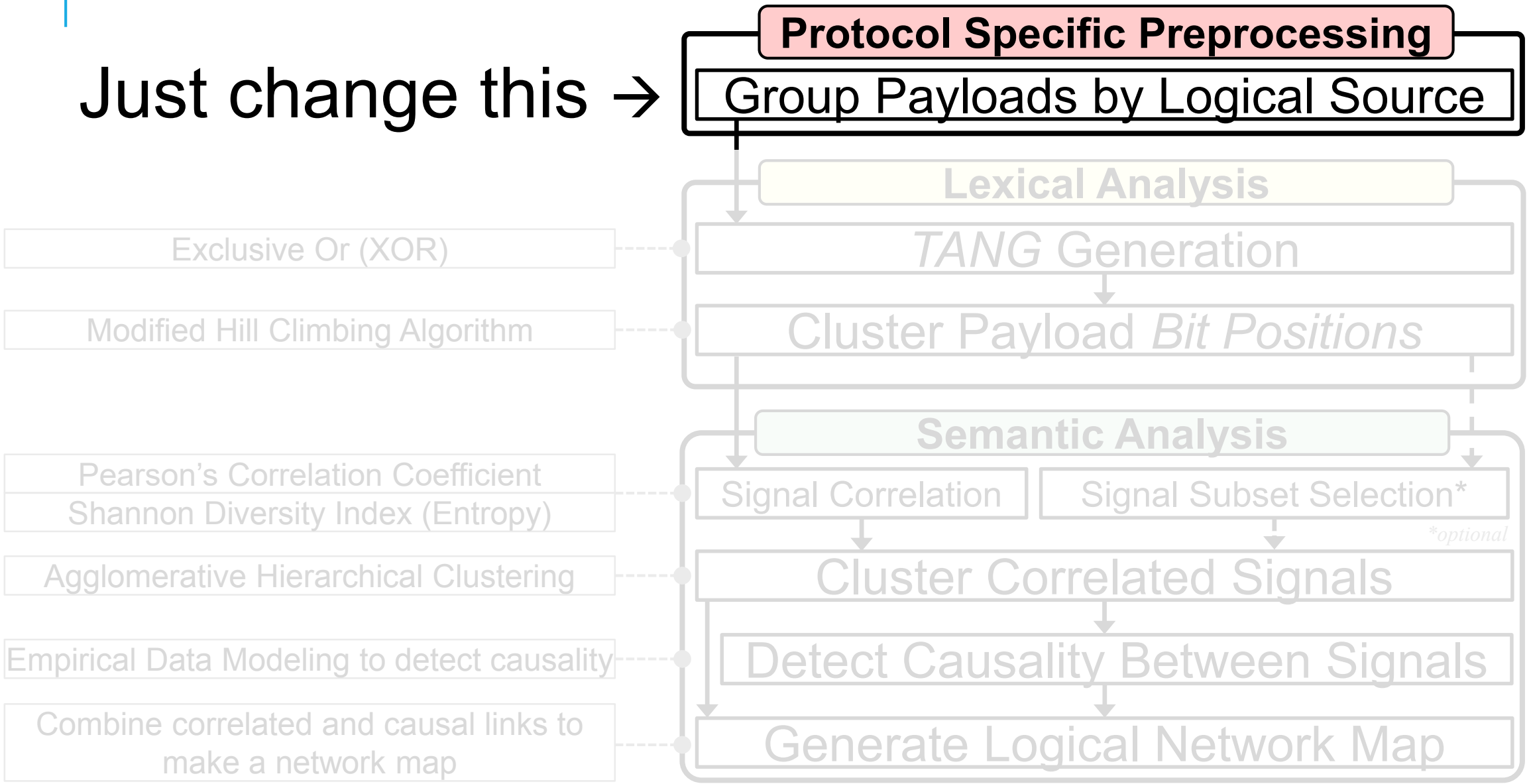


Code on GitHub does this…

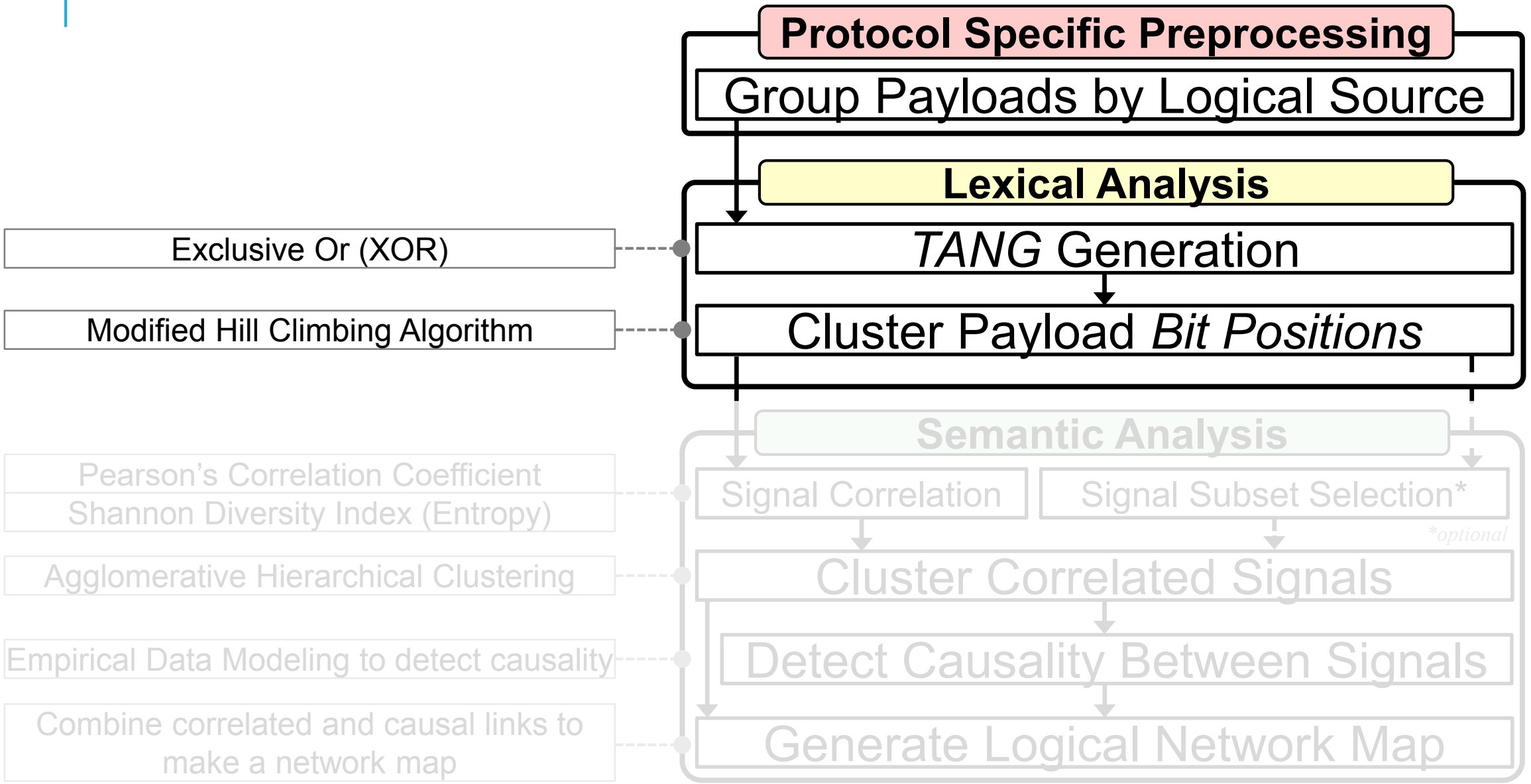


Different Control Network Protocol?

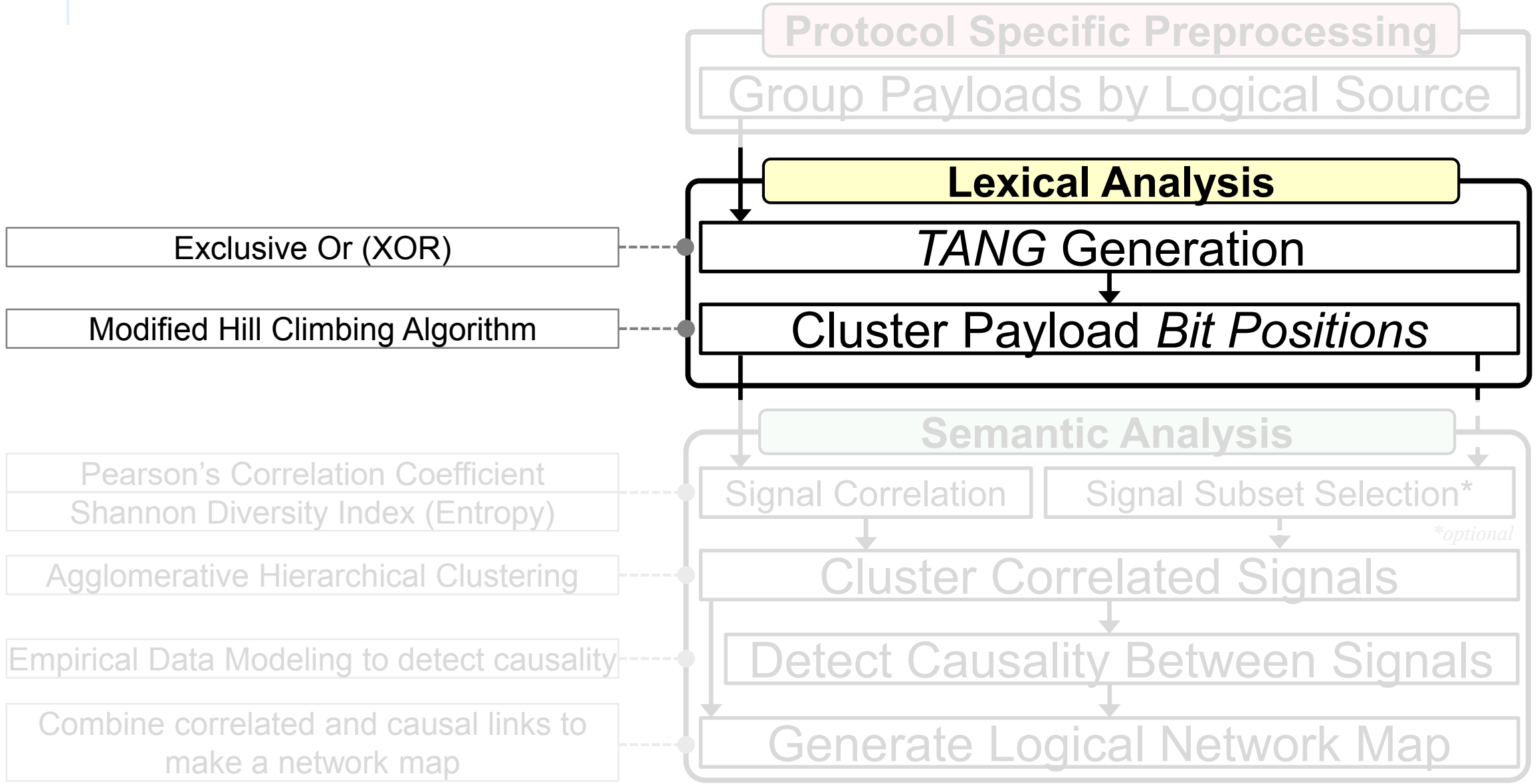
Just change this →



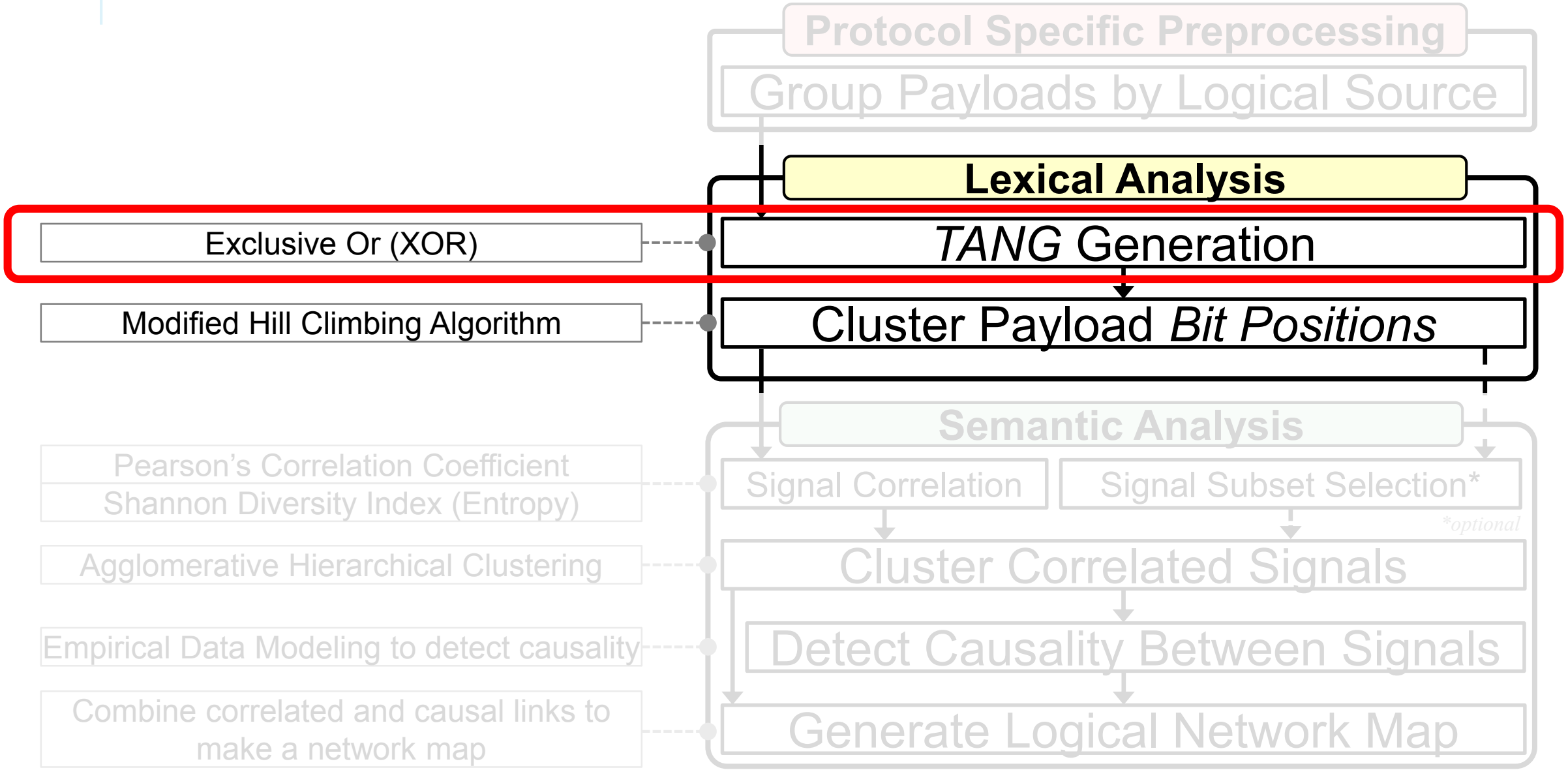
The demo is doing this…



I'll walk you through this…



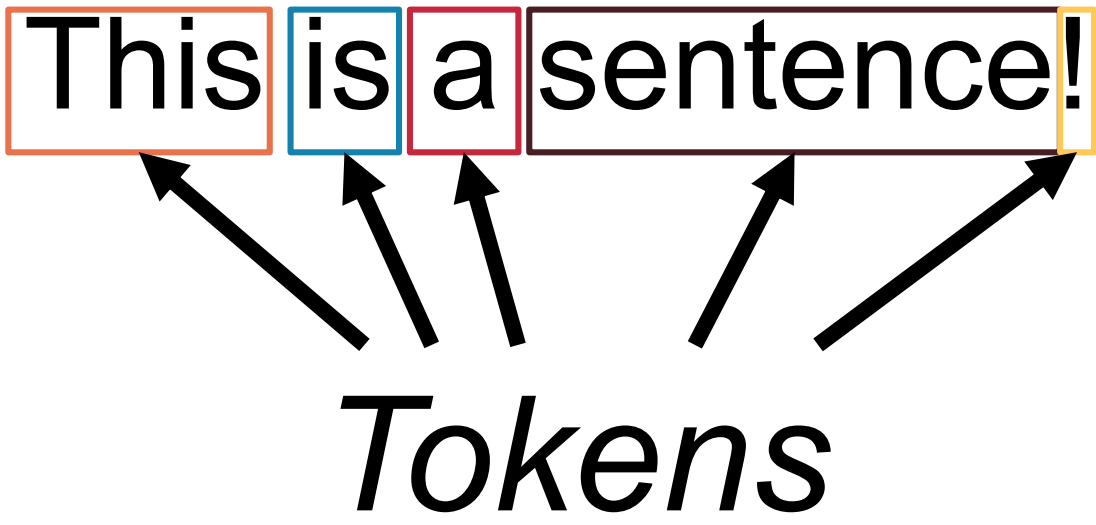
Unsupervised Reverse Engineering



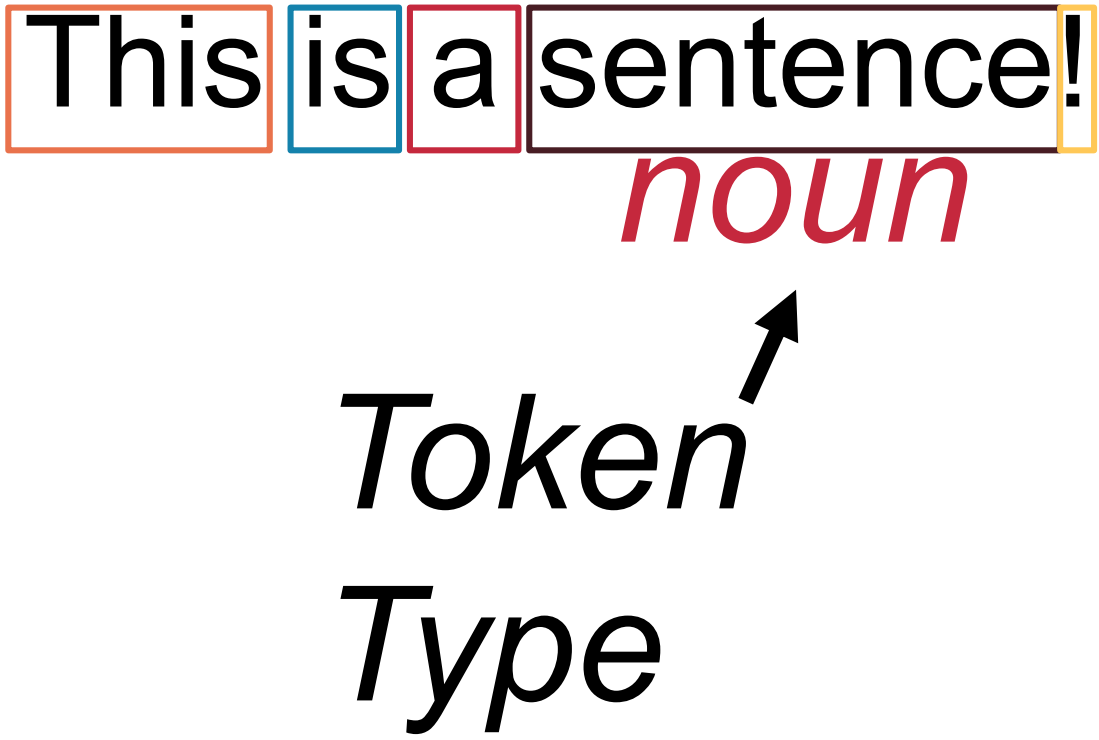
Lexical & Semantic Analysis

This is a sentence!

Lexical Analysis



Semantic Analysis



Lexical Analysis
Payload Tokenization

64-bit Payloads

Time	Bit 0	Bit 63
48.45	1	0
48.95	1	0
49.46	1	0
49.96	0	0
50.46	0	0
50.96	1	0
⋮	⋮ ⋮	⋮

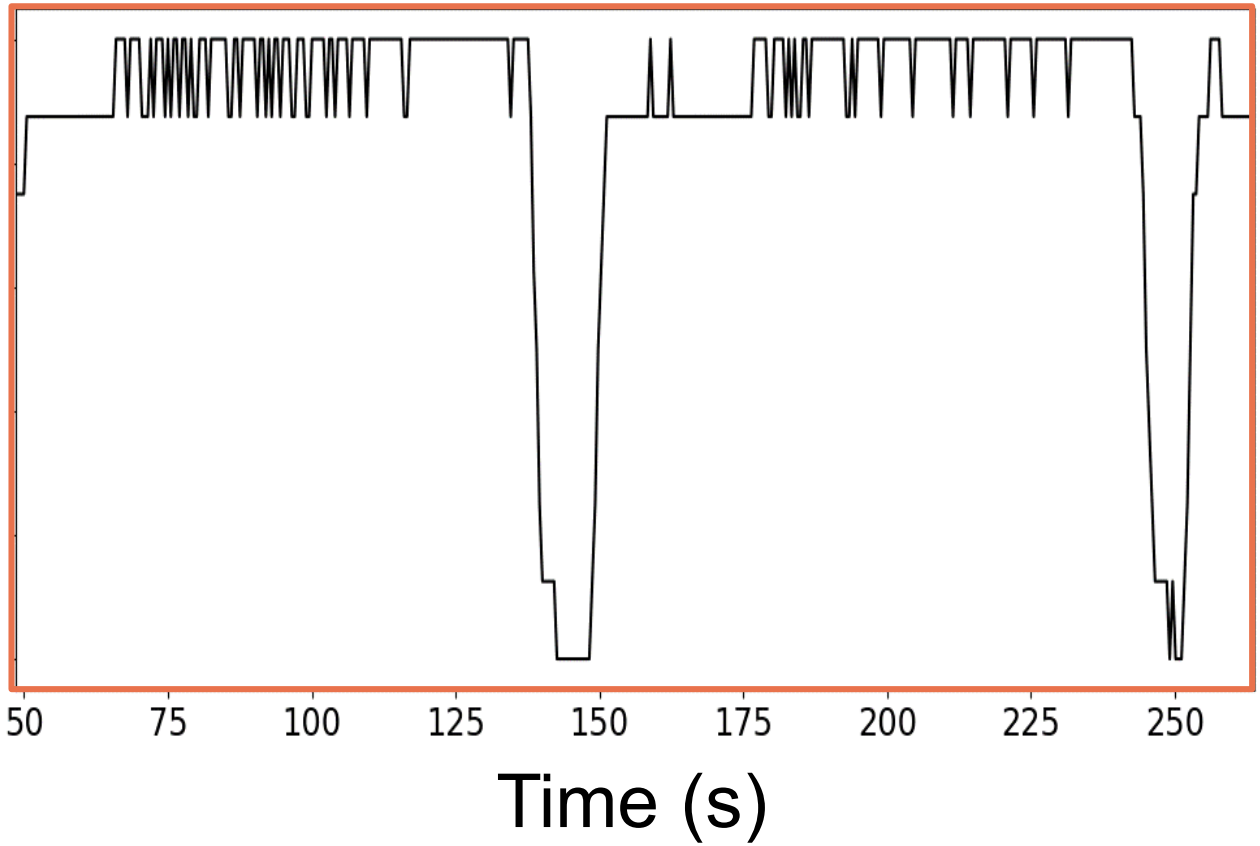
This is a sentence!

Lexical Analysis

Payload Tokenization

64-bit Payloads

Time	Bit 0	Bit 63
48.45	1	0
48.95	1	0
49.46	1	0
49.96	0	0
50.46	0	0
50.96	1	0
⋮	⋮ ⋮	⋮



Lexical Analysis

Payload Tokenization

64-bit Payloads

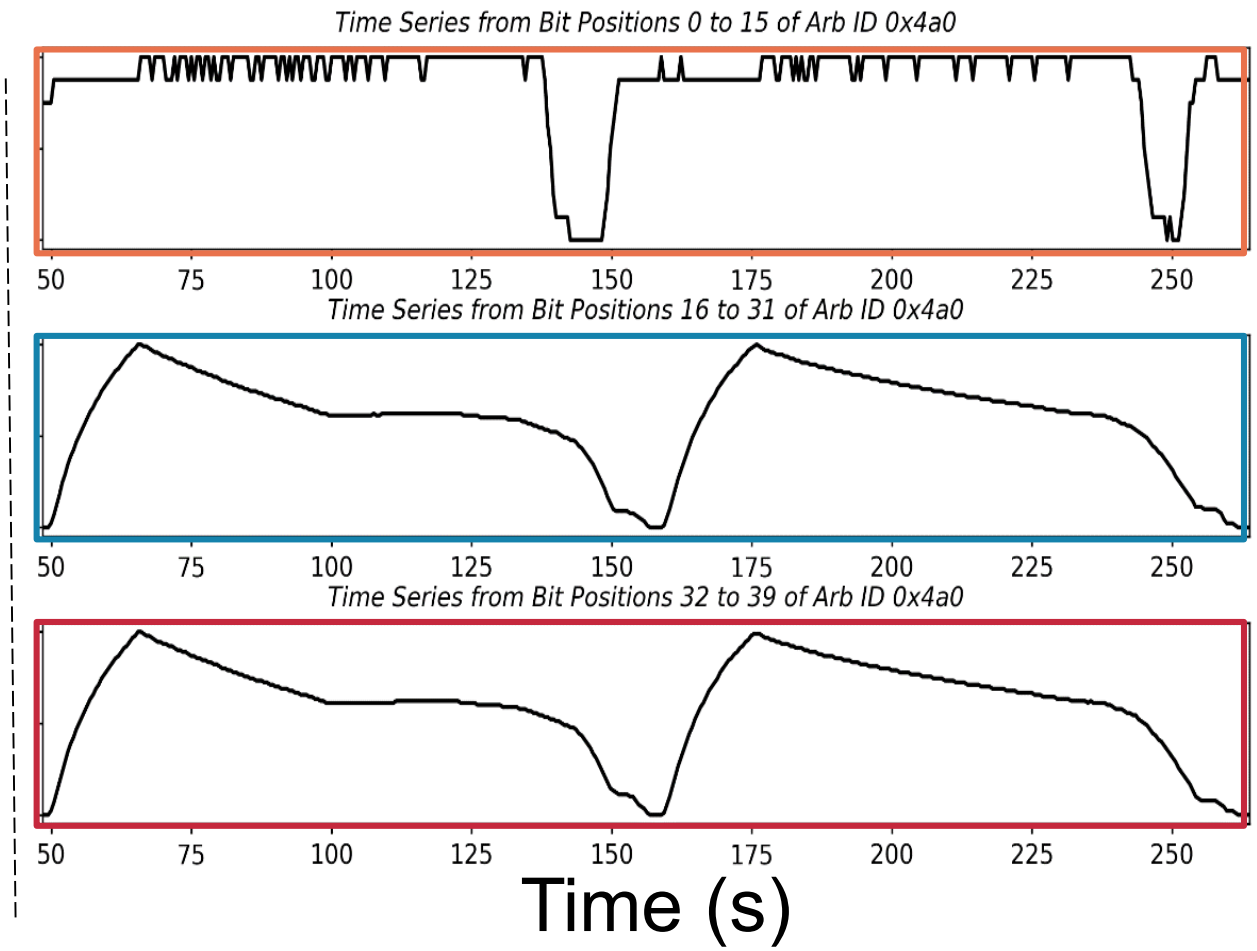
Time	Bit 0	Bit 63
48.45	1	0
48.95	1	0
49.46	1	0
49.96	0	0
50.46	0	0
50.96	1	0
⋮	⋮	⋮

This is a sentence!

Lexical Analysis
Payload Tokenization

64-bit Payloads

Time	Bit 0	Bit 63
48.45	1	0
48.95	1	0
49.46	1	0
49.96	0	0
50.46	0	0
50.96	1	0
⋮	⋮ ⋮	⋮



Payload Tokenization

By Least Significant Bit

Bit Position:	0	1	2	3	4	5	6	7	8	9
Observed Payloads	0	1	1	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	1
	1	0	0	1	0	0	0	0	1	0
	1	0	1	0	0	0	0	0	1	1
	1	0	1	1	0	0	0	1	0	0
	1	1	0	0	0	0	0	1	0	1
	1	1	0	1	0	0	0	1	1	0
	1	1	1	0	0	0	0	1	1	1

Payload Tokenization

By Least Significant Bit

Bit Position: 0 1 2 3 4 5 6 7 8 9

Exclusive-OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

0	1	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1	0	0
1	1	0	0	0	0	0	1	0	1
1	1	0	1	0	0	0	1	1	0

Payload Tokenization

By Least Significant Bit

Bit Position: 0 1 2 3 4 5 6 7 8 9

Exclusive-OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

1	1	1	1	0	0	0	0	0	1
0	0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	0	0	0	1
0	0	0	1	0	0	0	1	1	1
0	1	1	1	0	0	0	0	0	1
0	0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	0	0	0	1

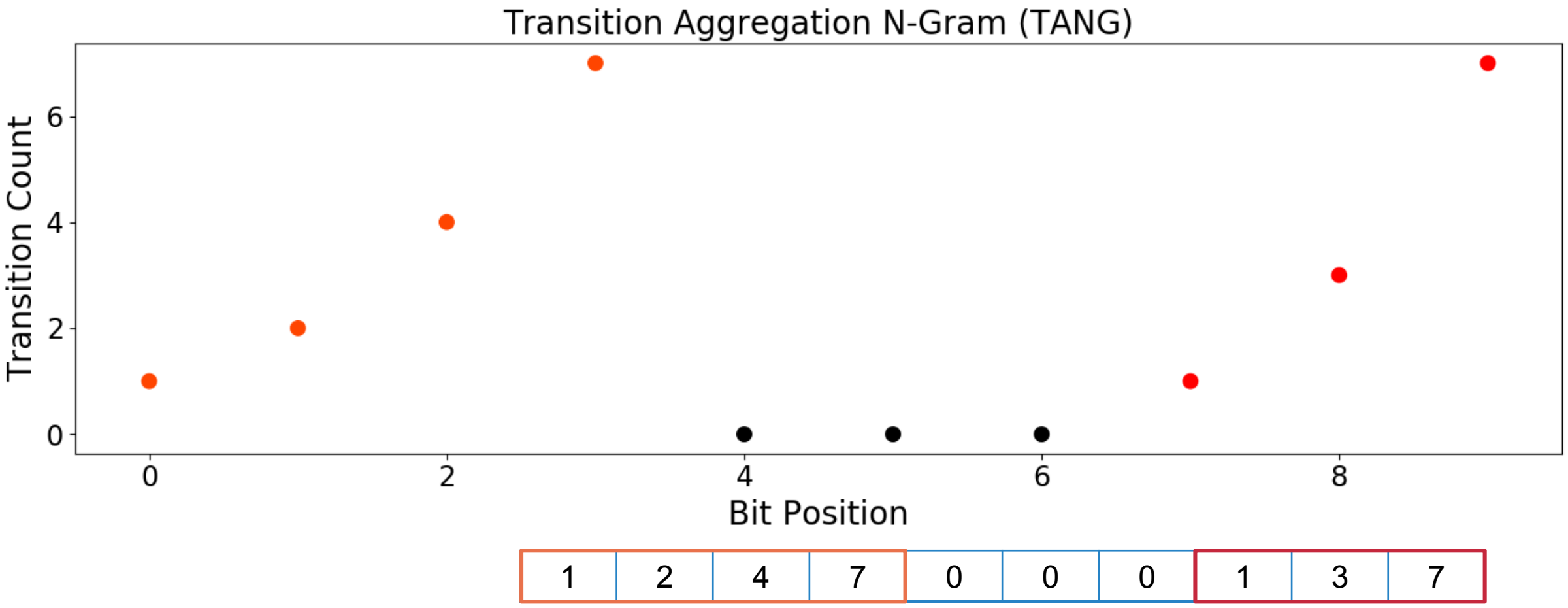
Payload Tokenization

By Least Significant Bit

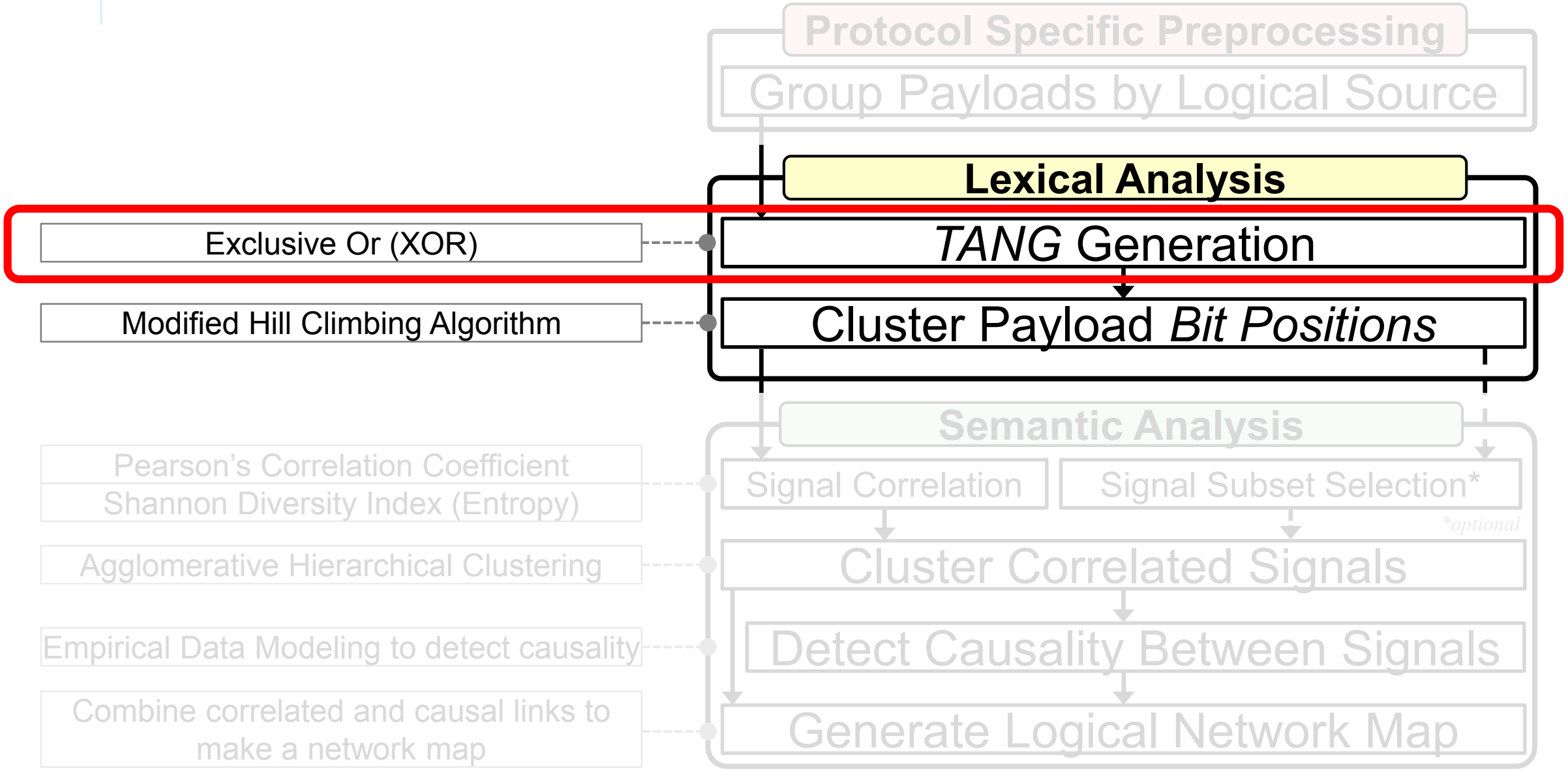
Bit Position:	0	1	2	3	4	5	6	7	8	9
	1	1	1	1	0	0	0	0	0	1
	0	0	0	1	0	0	0	0	1	1
	0	0	1	1	0	0	0	0	0	1
	0	0	0	1	0	0	0	1	1	1
	0	1	1	1	0	0	0	0	0	1
	0	0	0	1	0	0	0	0	1	1
+	0	0	1	1	0	0	0	0	0	1
	1	2	4	7	0	0	0	1	3	7

Payload Tokenization

By Least Significant Bit

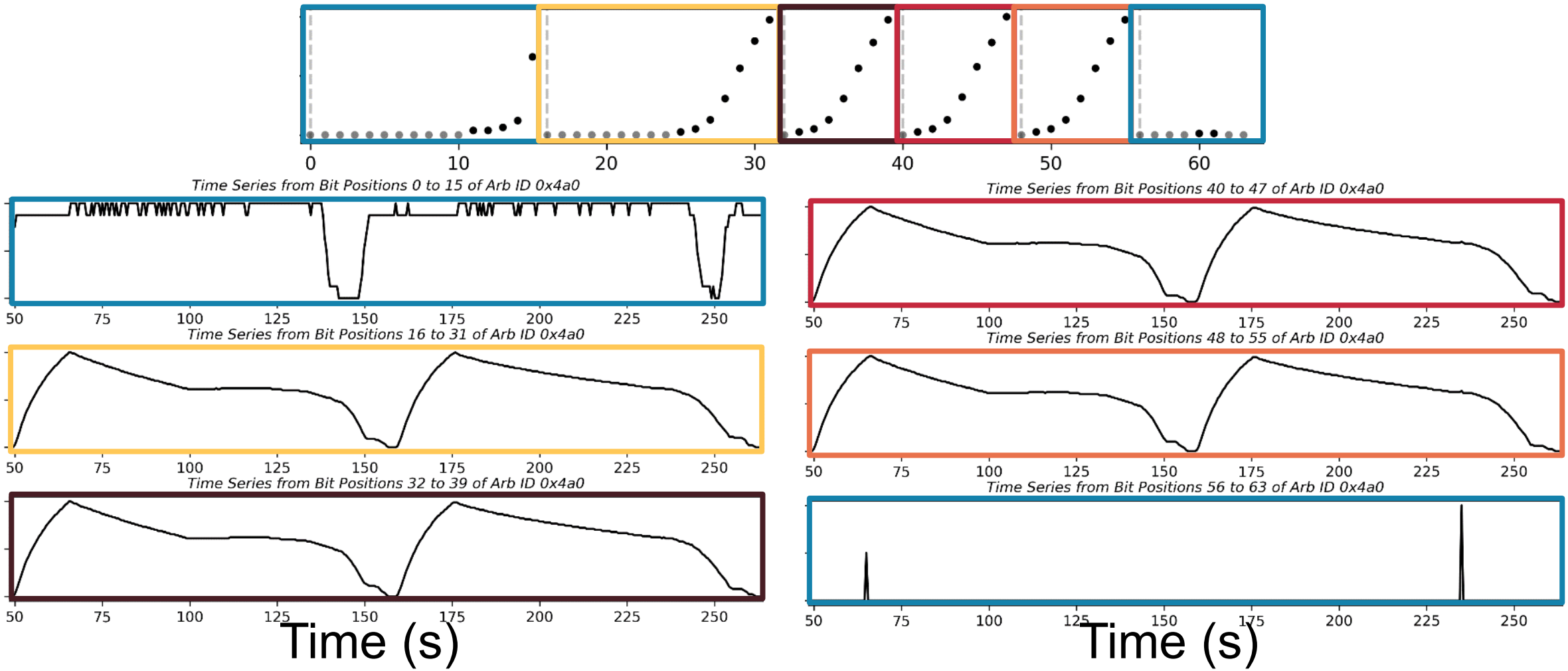


Unsupervised Reverse Engineering

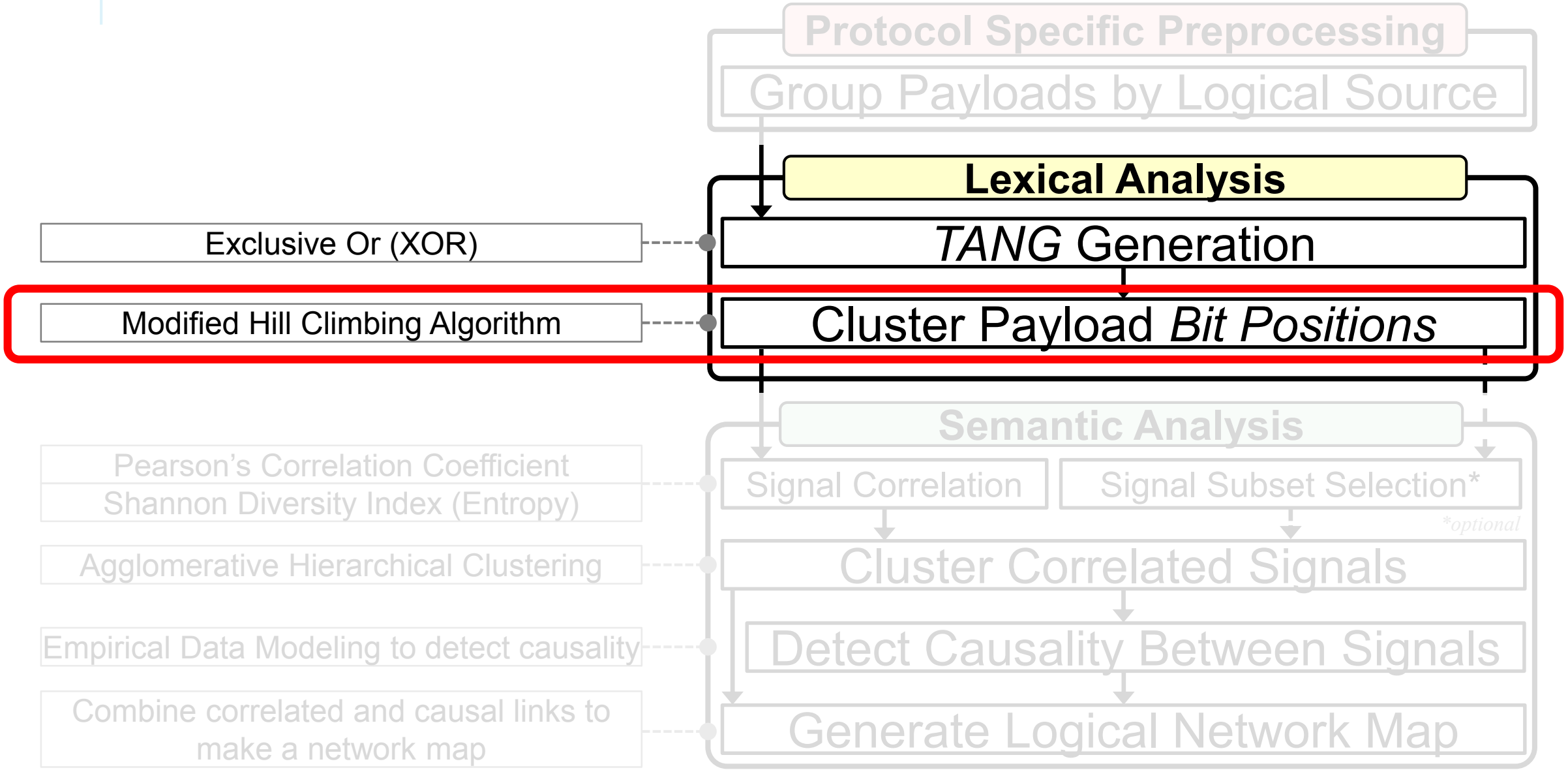


Payload Tokenization

By Least Significant Bit

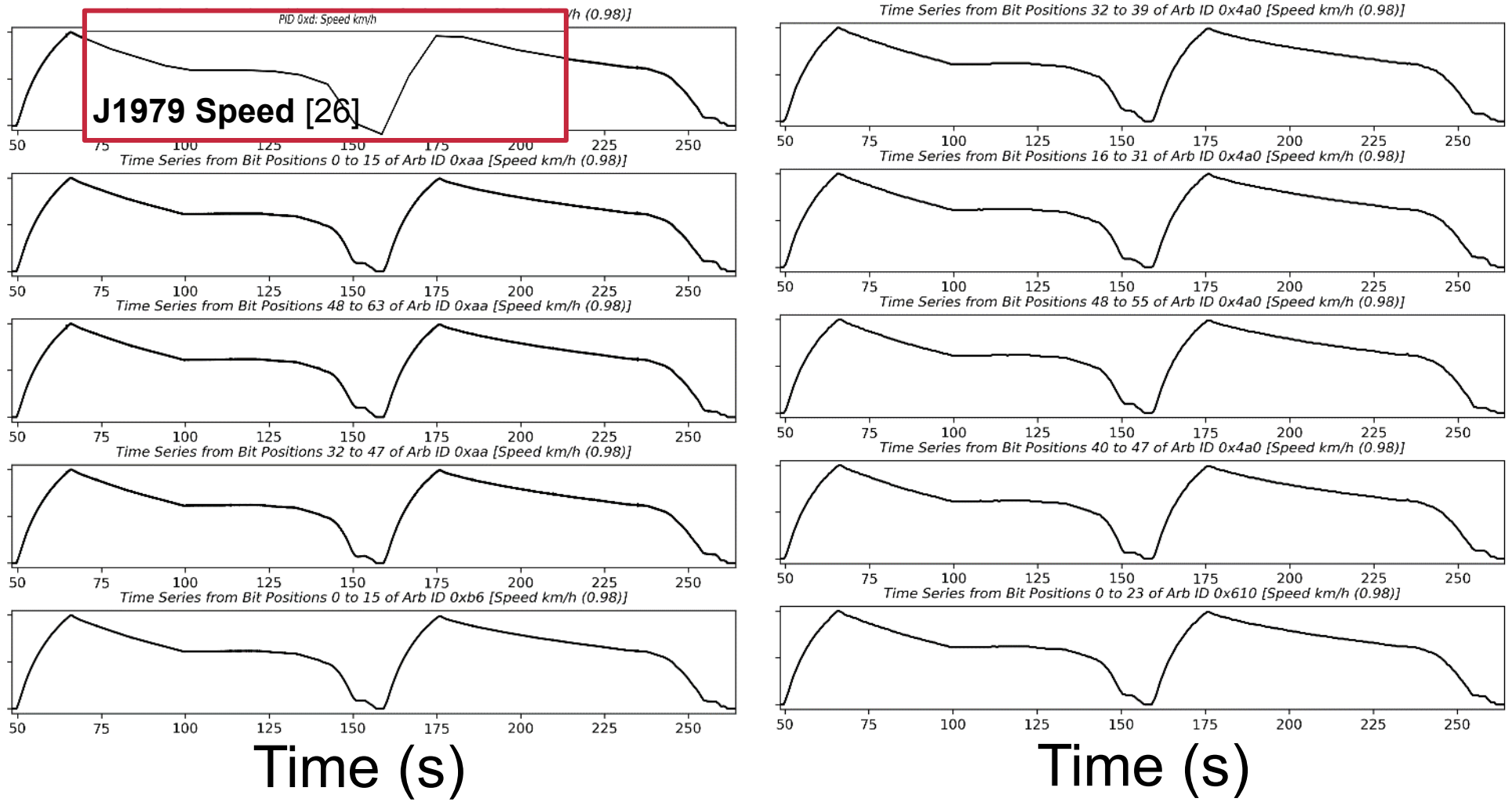


Unsupervised Reverse Engineering



Semantic Analysis

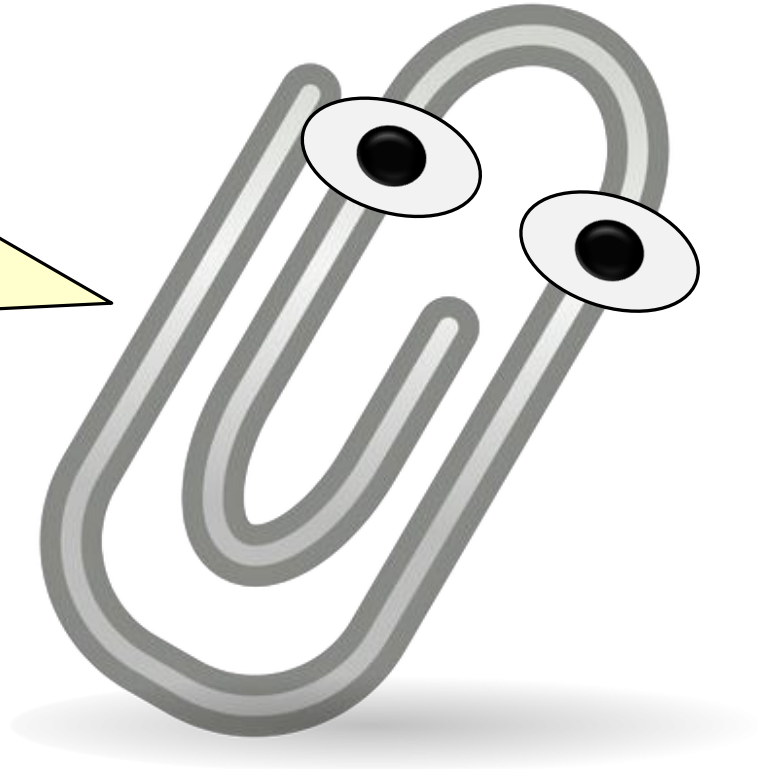
Correlated and Causal Relationships



[26] SAE International, “SAE J1979: E/E Diagnostic Test Modes,” 2017.

SHOW ME WHAT YOU GOT!

Let’s reverse engineer some cars!

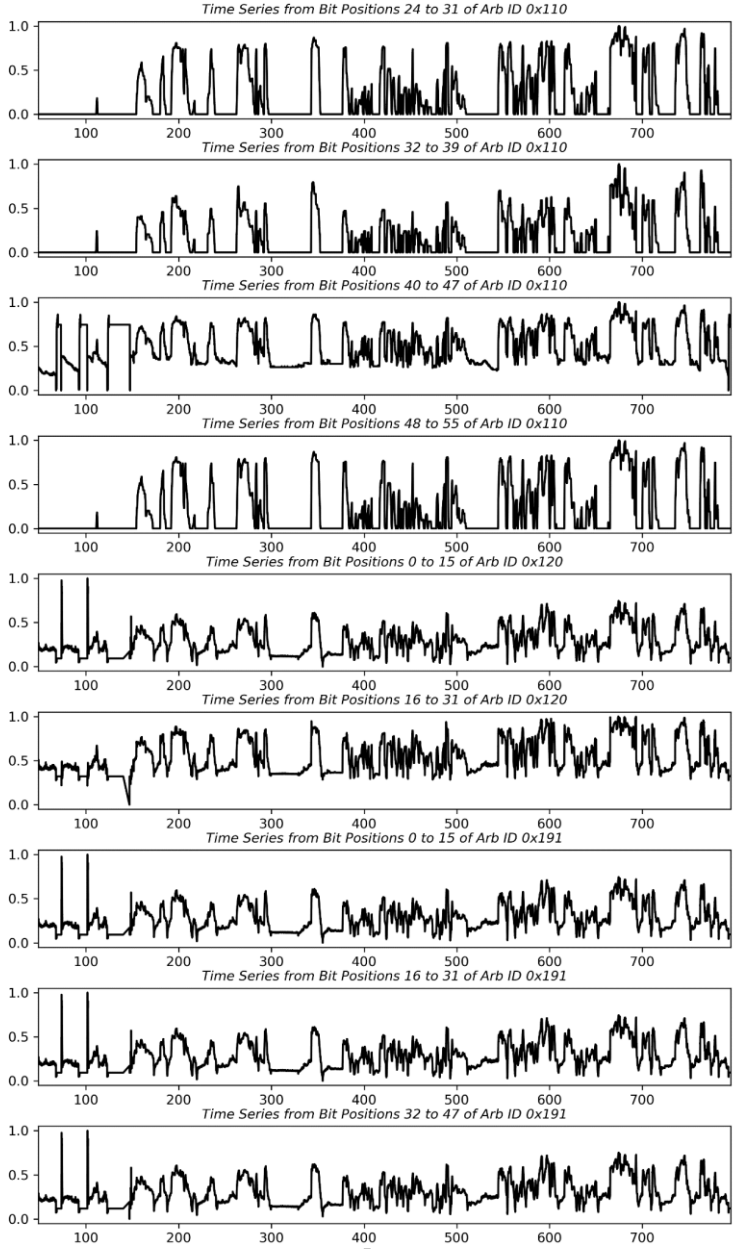


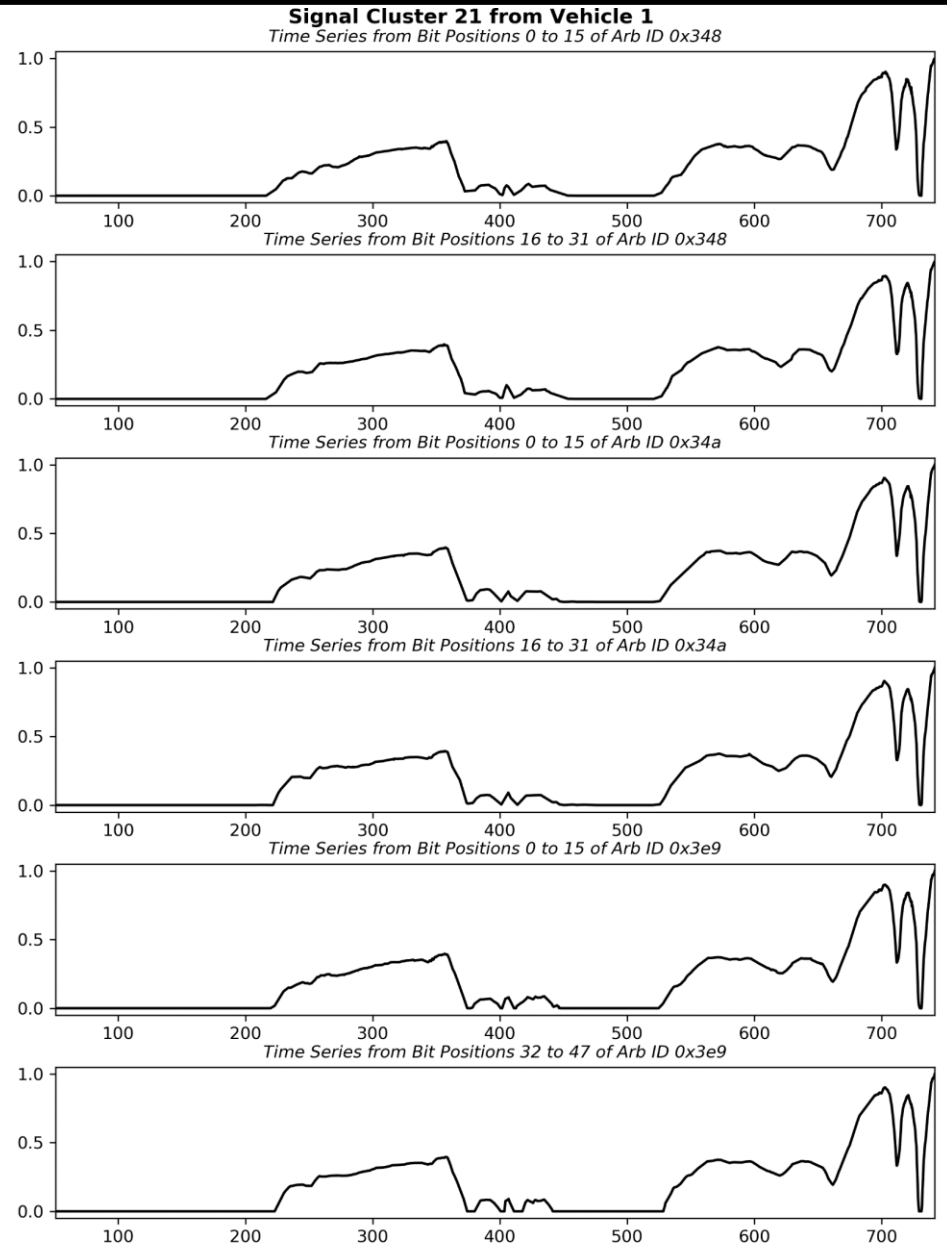
https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 1

VEHICLE 2

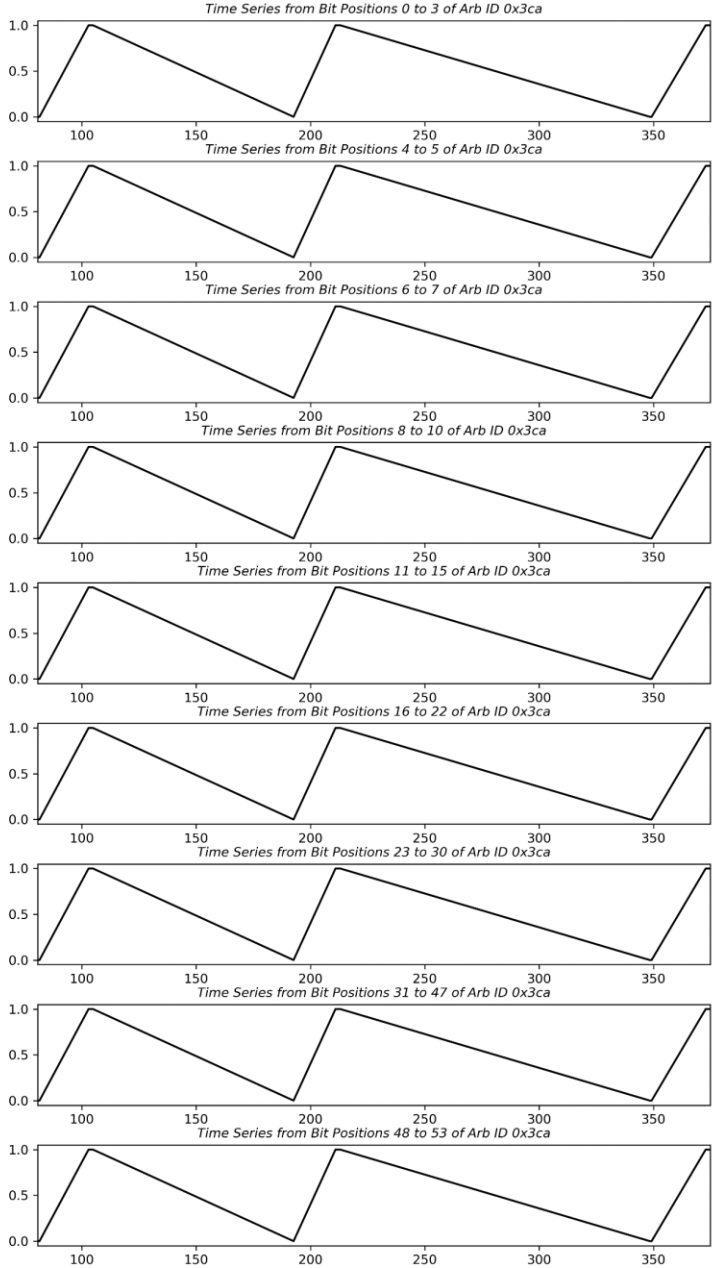
CROPPED TO FIT ON SLIDE





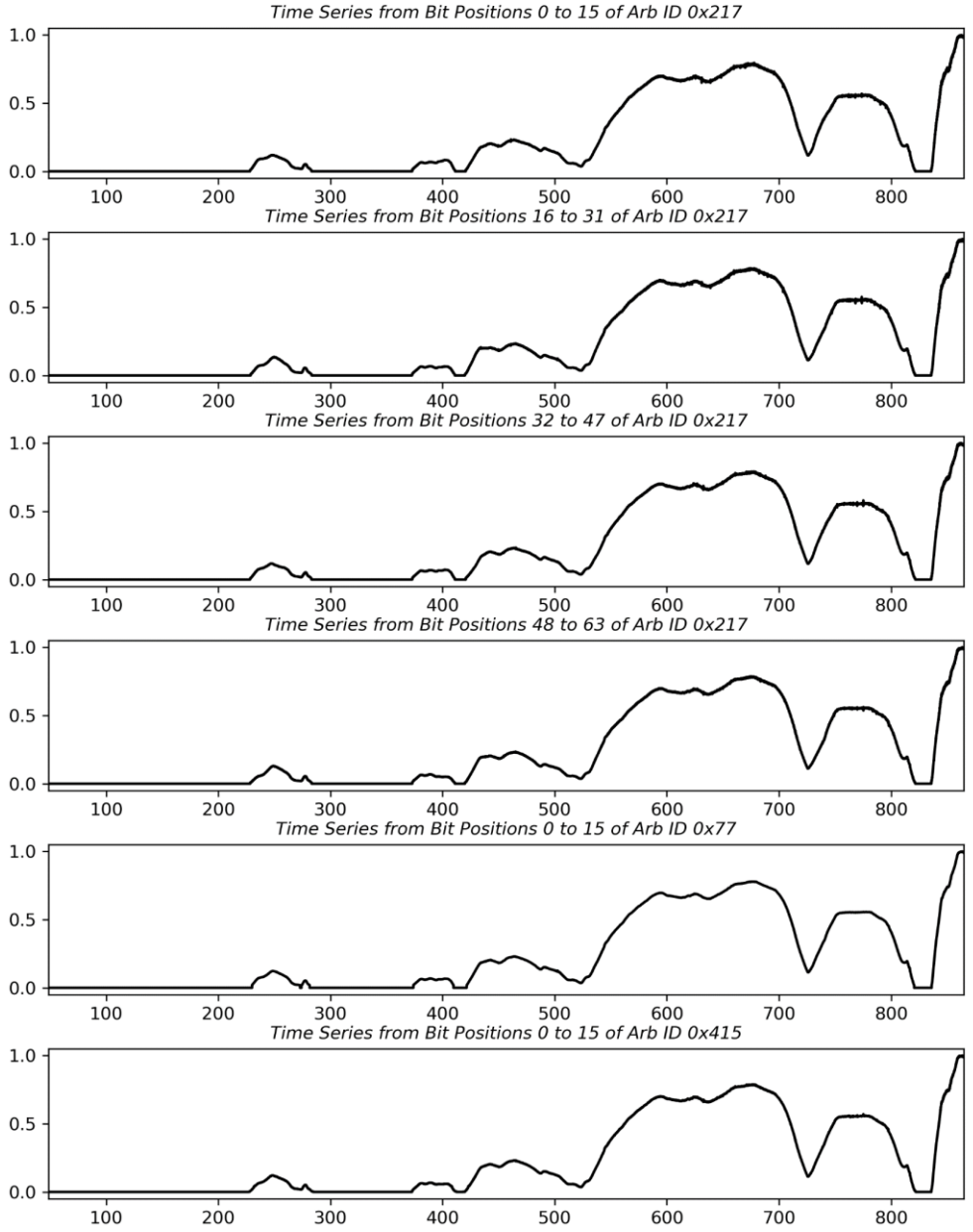
https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 3



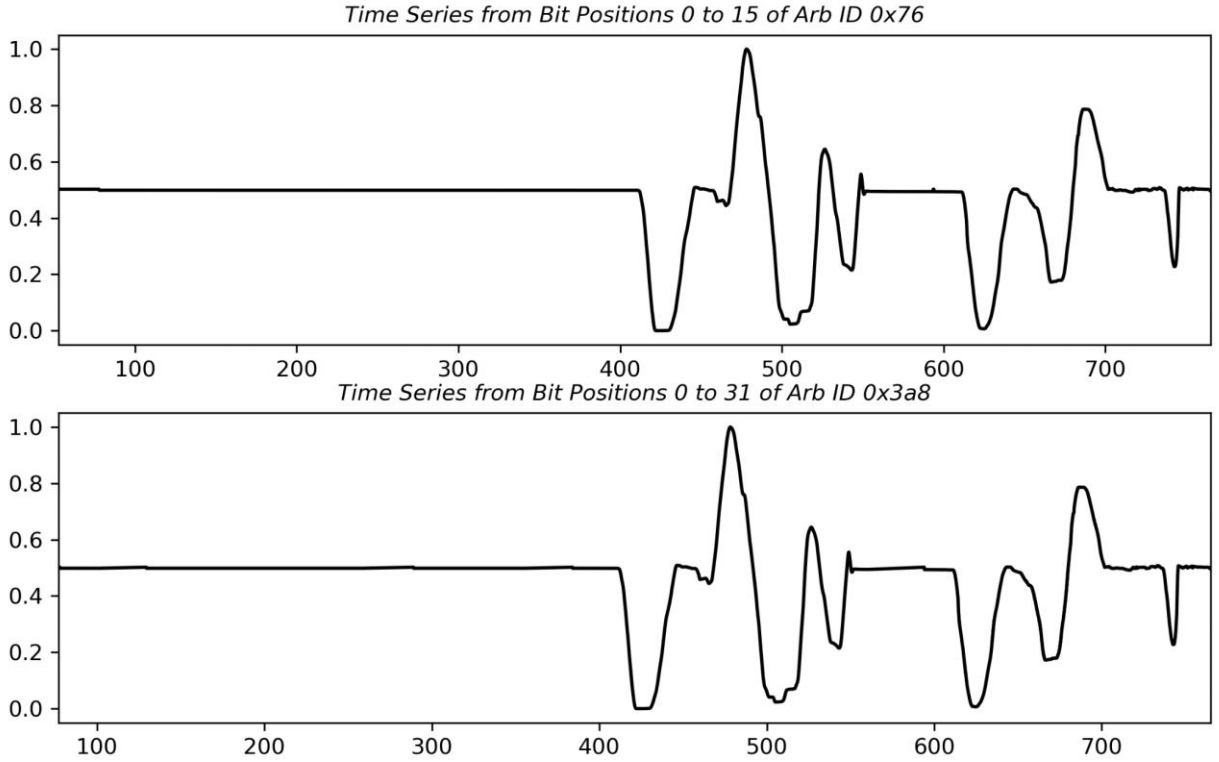
CROPPED TO FIT ON SLIDE

VEHICLE 4

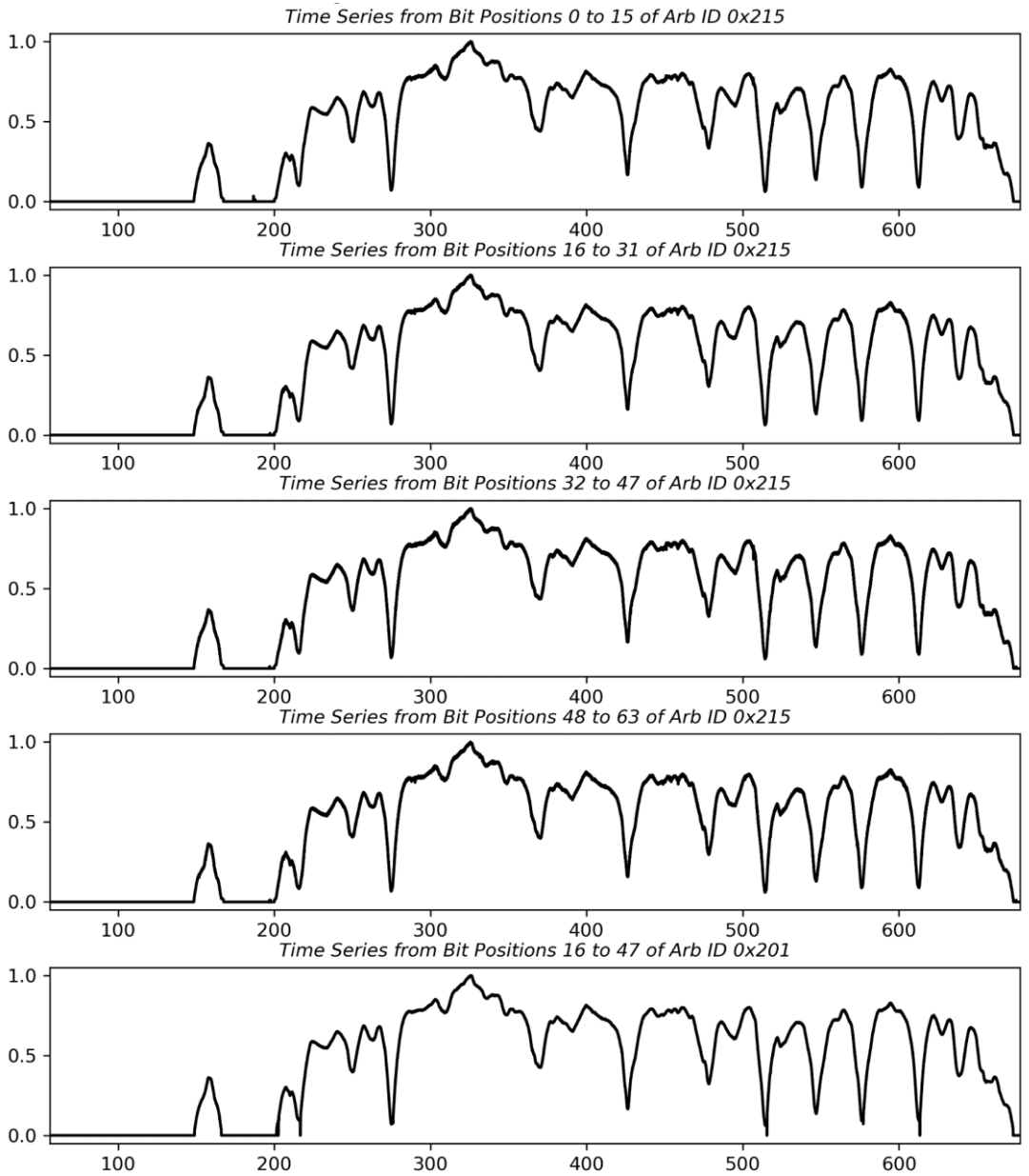


https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 5

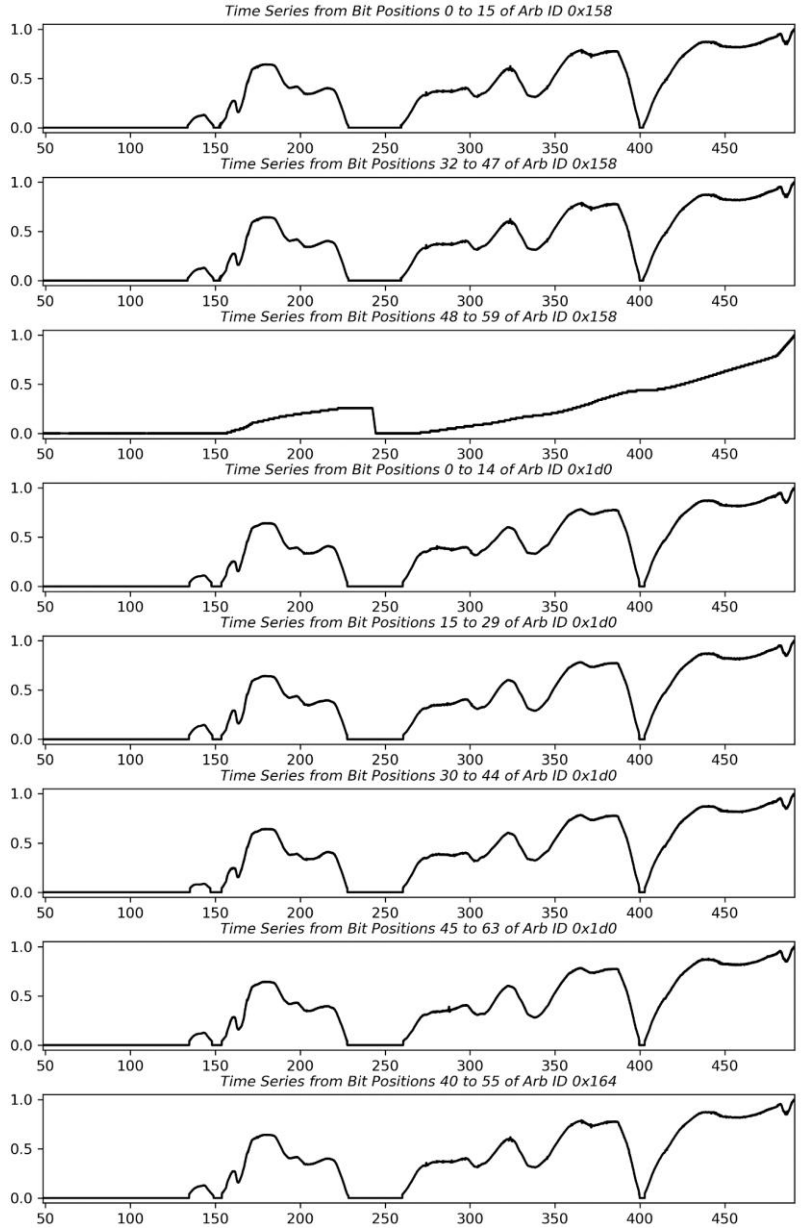


VEHICLE 6



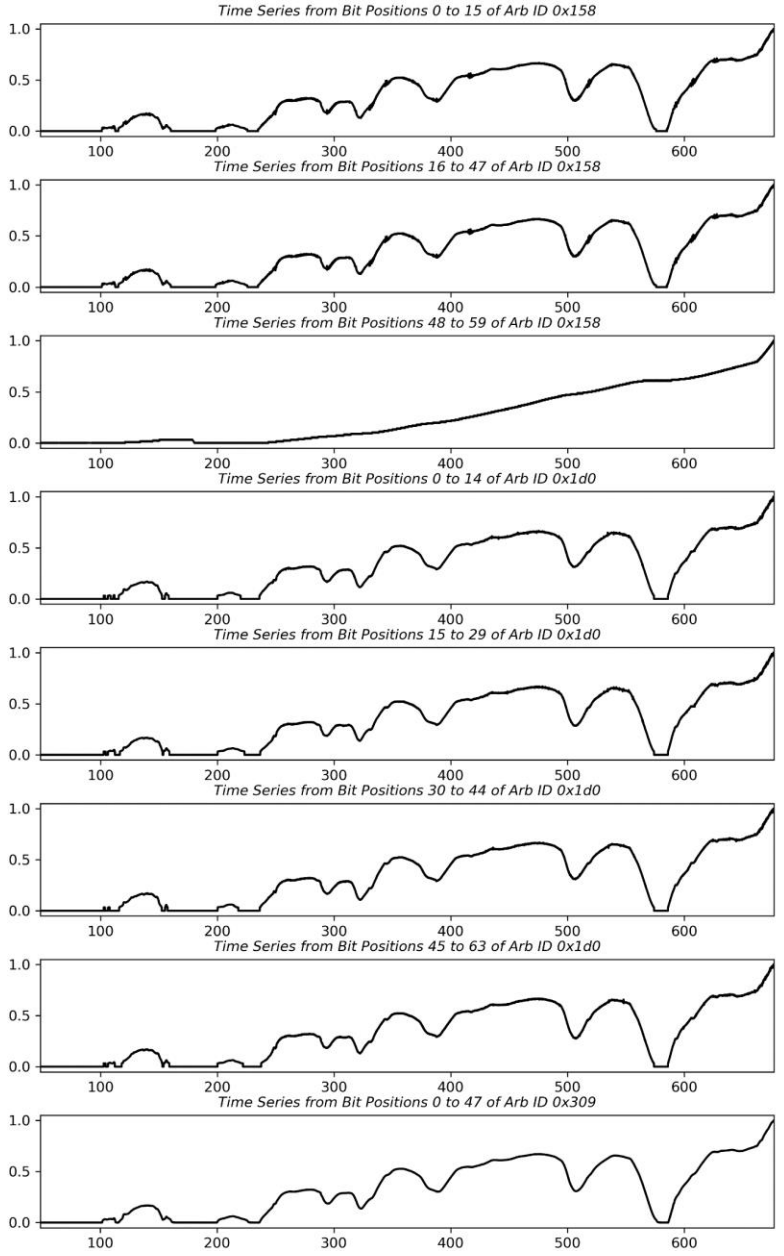
https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 7



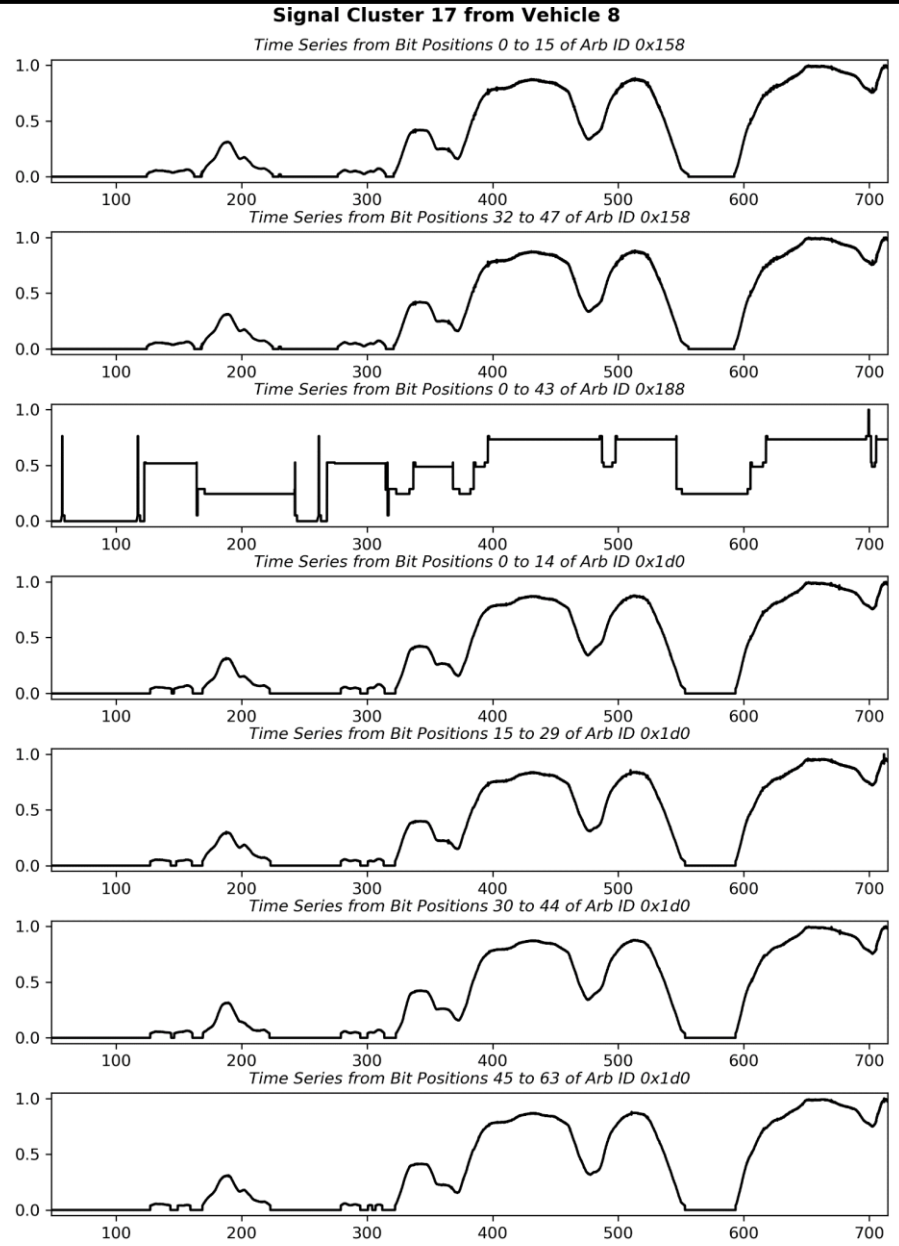
CROPPED TO FIT ON SLIDE

VEHICLE 8



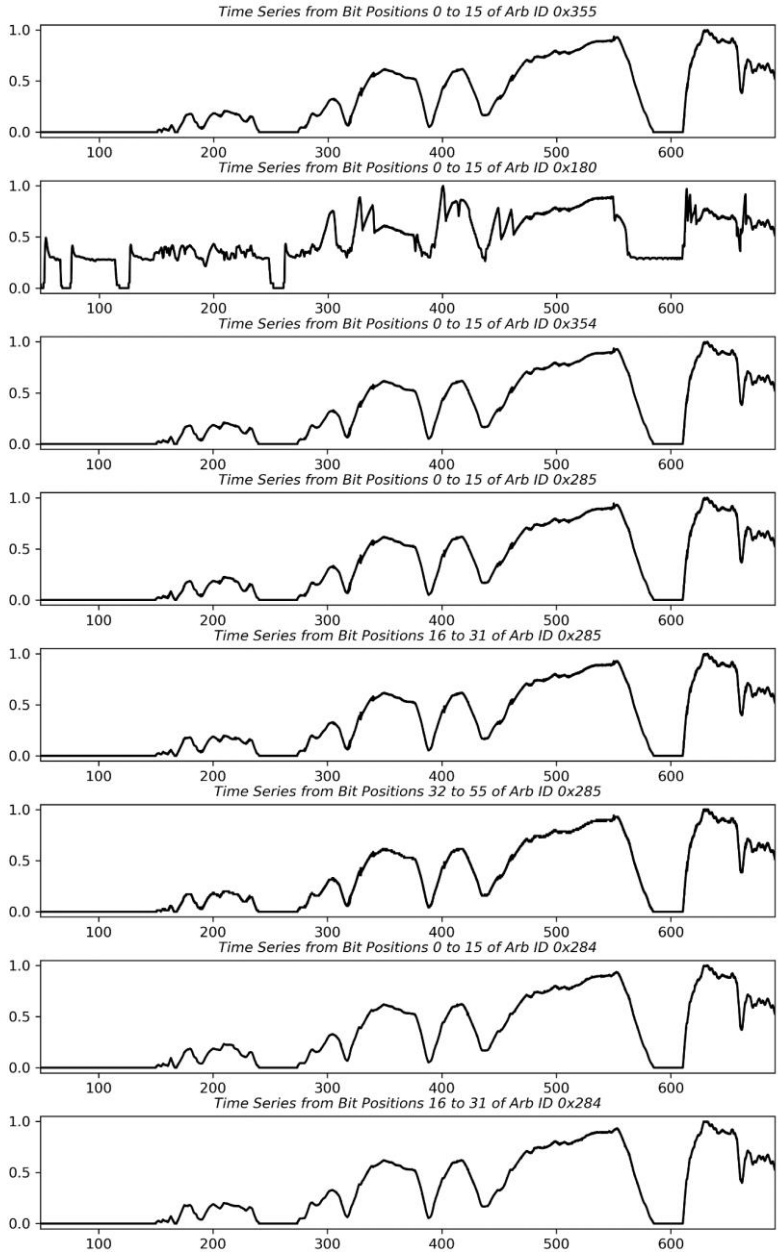
https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 9



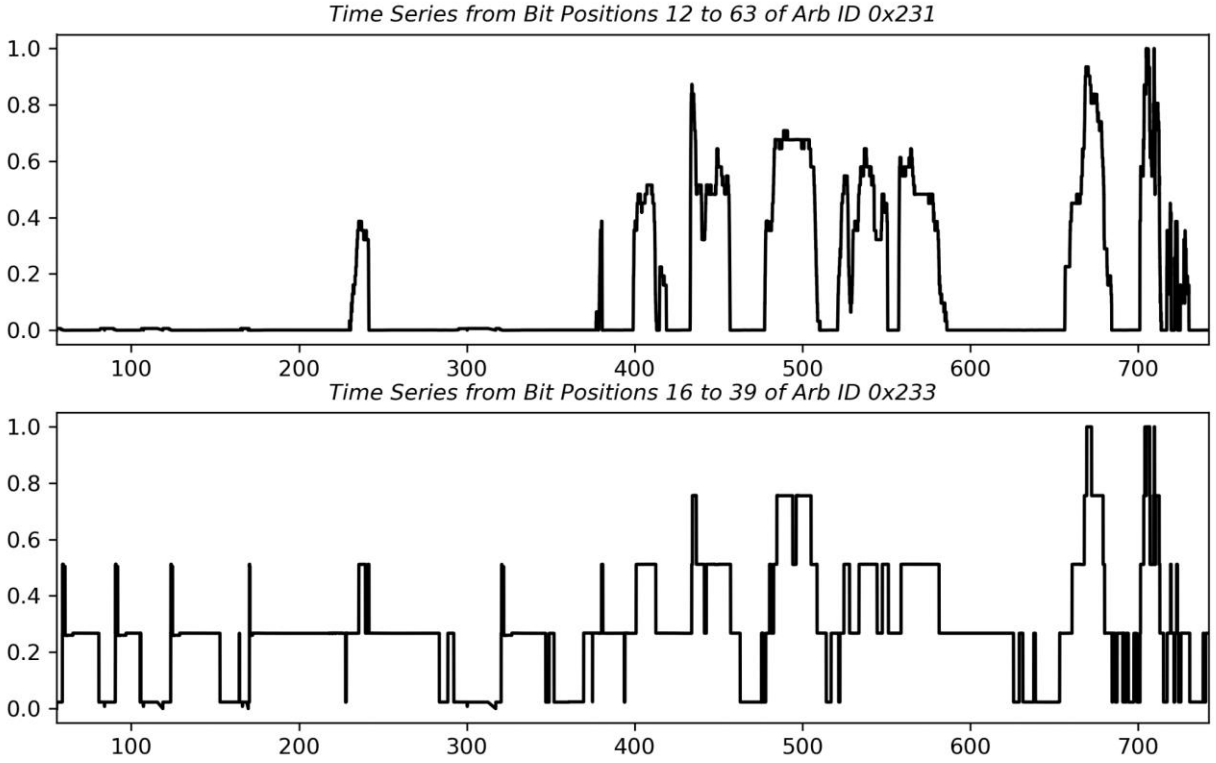
CROPPED TO FIT ON SLIDE

VEHICLE 10

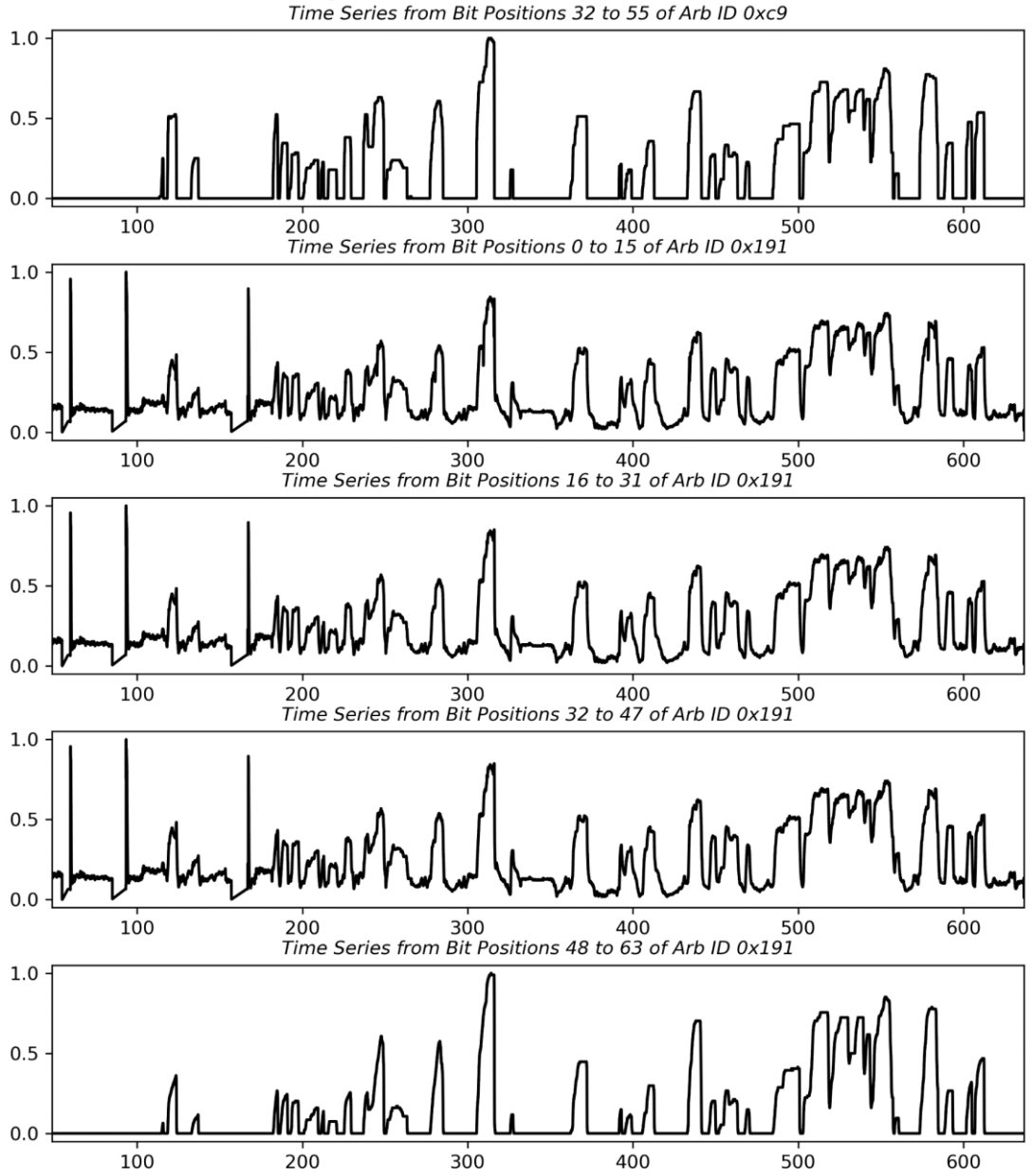


https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 11

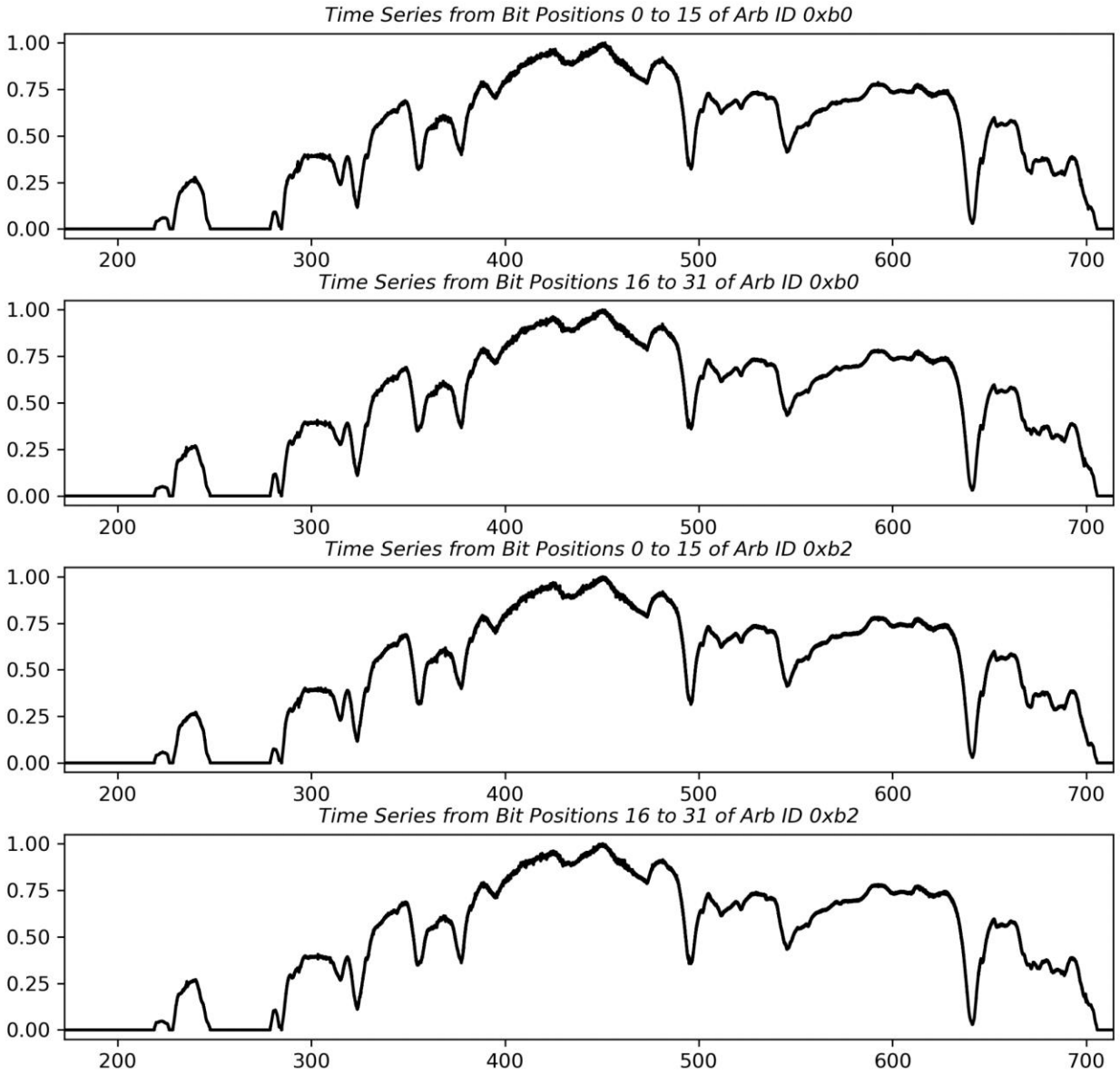


VEHICLE 12

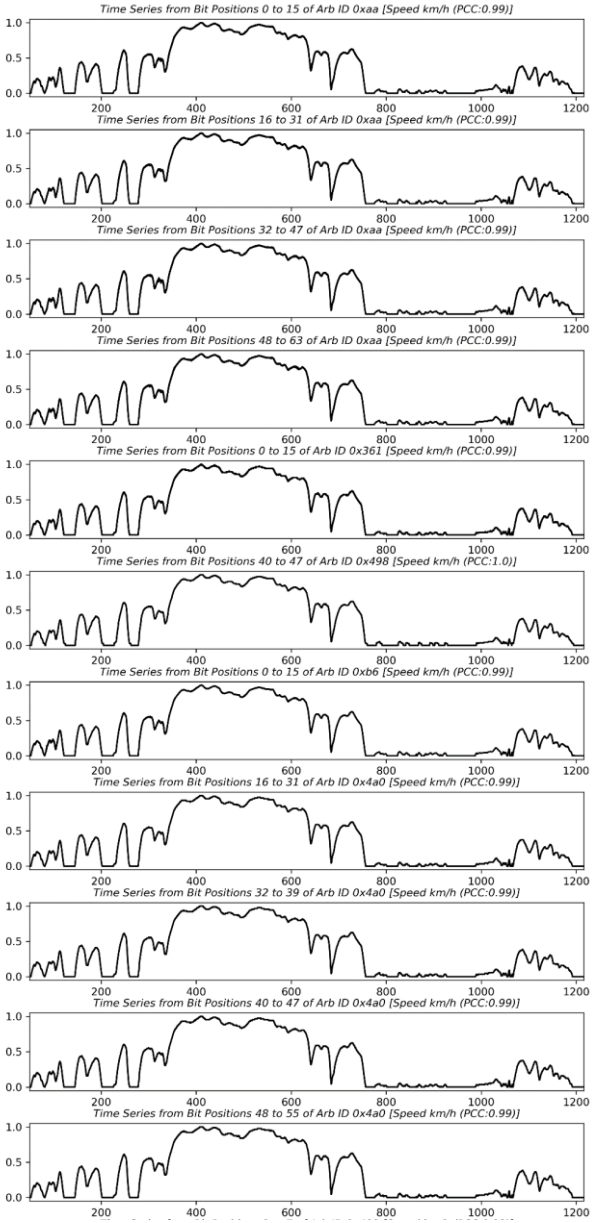


https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 13



VEHICLE 14



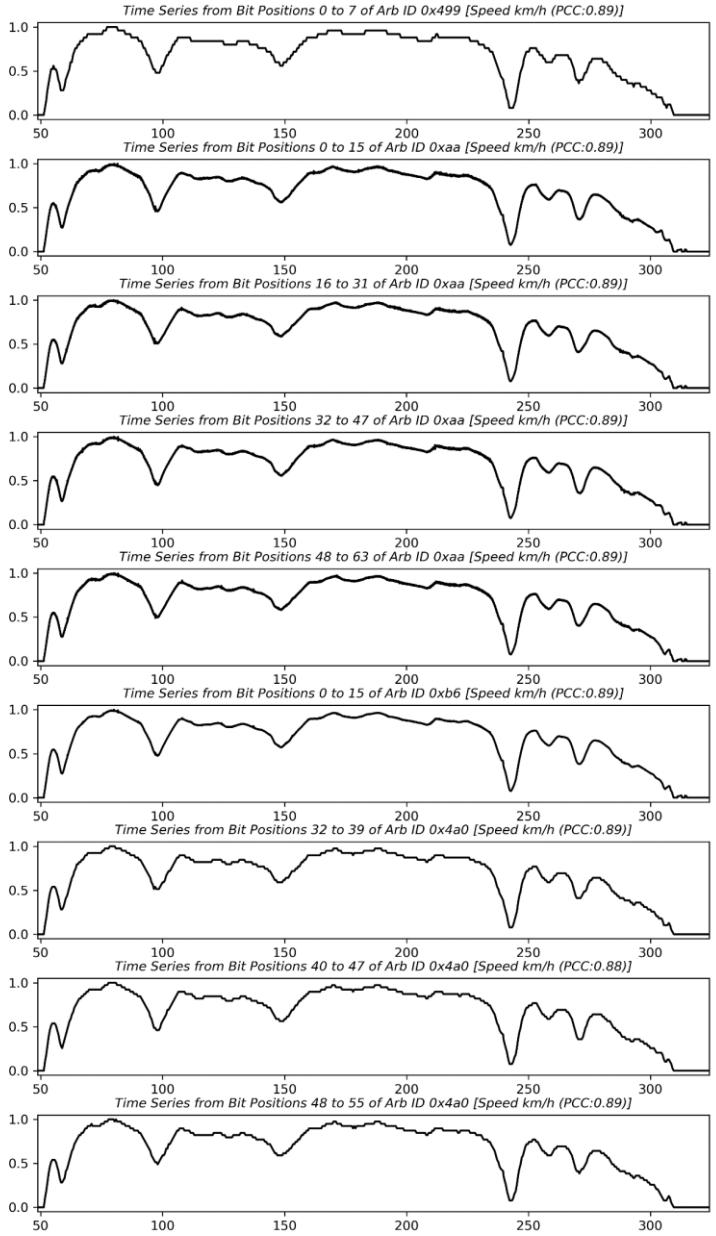
CROPPED TO FIT ON SLIDE

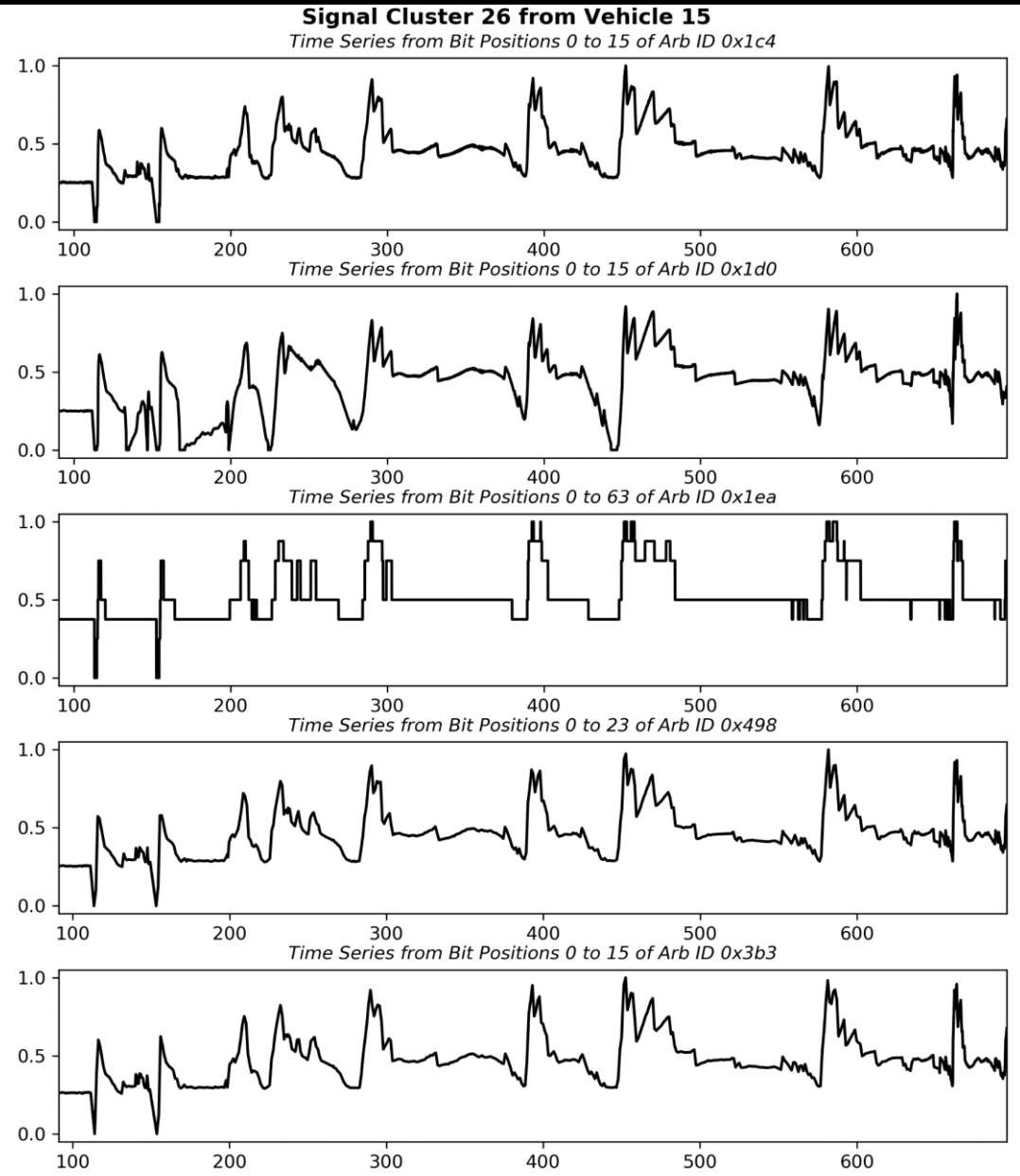
https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 15

VEHICLE 16

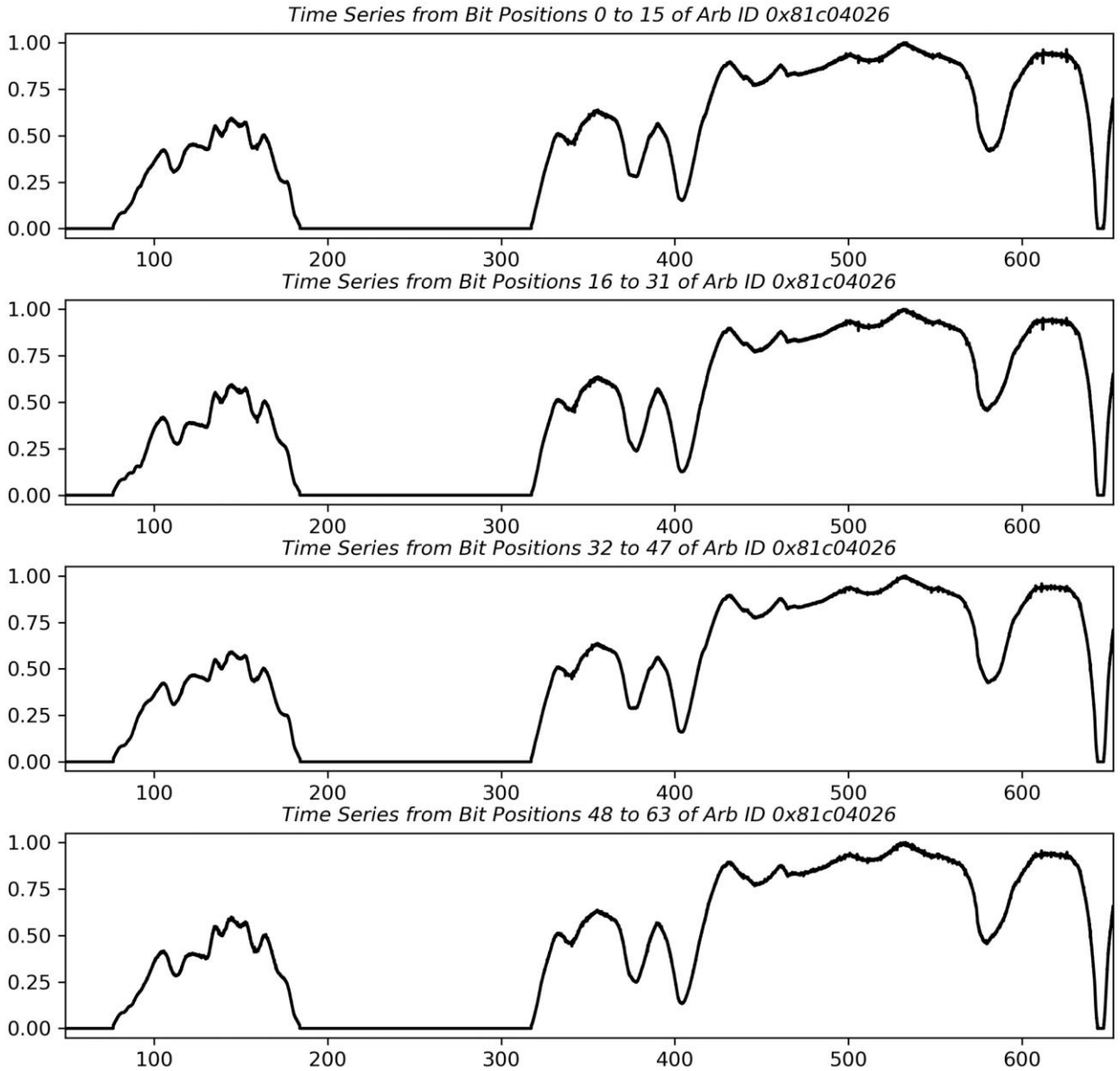
CROPPED TO FIT ON SLIDE



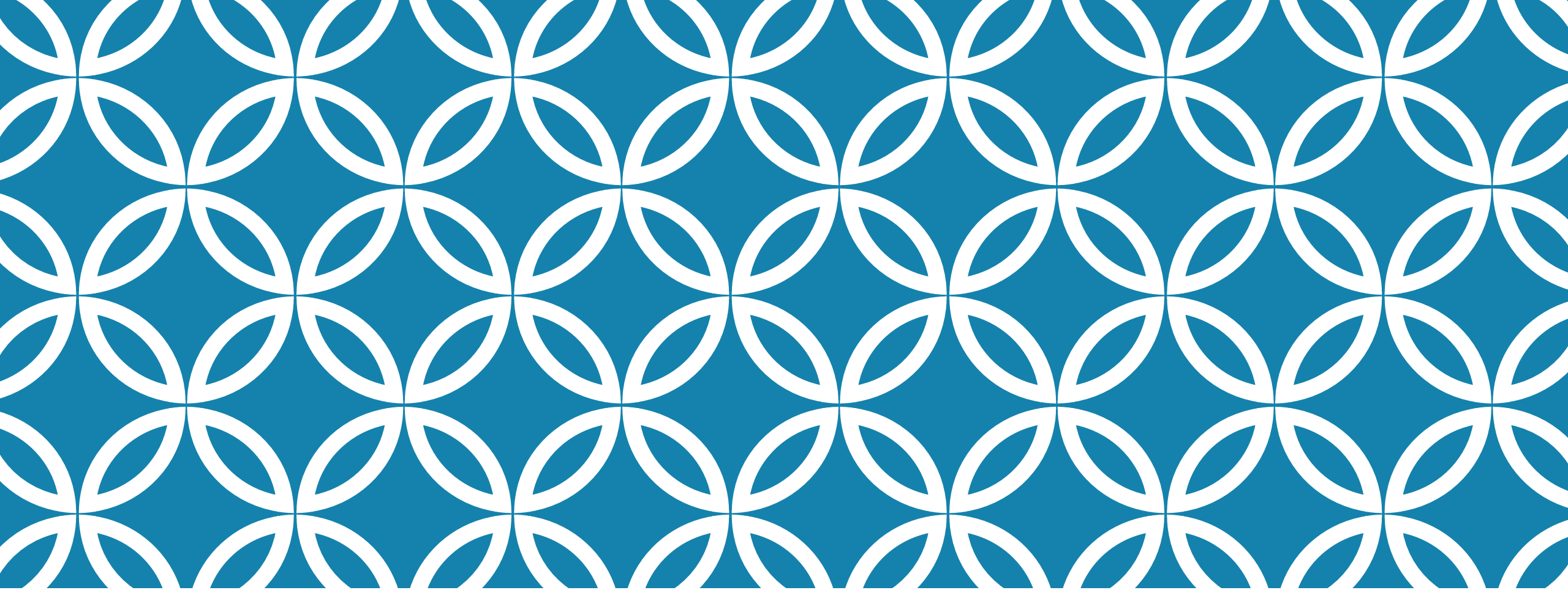


https://github.com/brent-stone/CAN_Reverse_Engineering

VEHICLE 17



https://github.com/brent-stone/CAN_Reverse_Engineering



QUESTIONS

BRENT STONE