Focusing and Calibration of Large Scale Network Sensors using GraphBLAS Anonymized Hypersparse Matrices

Jeremy Kepner¹, Michael Jones¹, Phil Dykstra², Chansup Byun¹, Timothy Davis³, Hayden Jananthan¹, William Arcand¹, David Bestor¹, William Bergeron¹, Vijay Gadepally¹, Micheal Houle¹, Matthew Hubbell¹, Anna Klein¹, Lauren Milechin¹, Guillermo Morales¹, Julie Mullen¹, Ritesh Patel¹, Alex Pentland¹, Sandeep Pisharody¹, Andrew Prout¹, Albert Reuther¹, Antonio Rosa¹, Siddharth Samsi¹, Tyler Trigg¹, Charles Yee¹, Peter Michaleas¹

¹MIT, ²HPCMP DREN, ³Texas A&M,

Abstract—Defending community-owned cyber space requires community-based efforts. Large-scale network observations that uphold the highest regard for privacy are key to protecting our shared cyberspace. Deployment of the necessary network sensors requires careful sensor placement, focusing, and calibration with significant volumes of network observations. This paper demonstrates novel focusing and calibration procedures on a multi-billion packet dataset using high-performance GraphBLAS anonymized hypersparse matrices. The run-time performance on a real-world data set confirms previously observed realtime processing rates for high-bandwidth links while achieving significant data compression. The output of the analysis demonstrates the effectiveness of these procedures at focusing the traffic matrix and revealing the underlying stable heavy-tail statistical distributions that are necessary for anomaly detection. A simple model of the corresponding probability of detection (p_d) and probability of false alarm (p_{fa}) for these distributions highlights the criticality of network sensor focusing and calibration. Once a sensor is properly focused and calibrated it is then in a position to carry out two of the central tenets of good cybersecurity: (1) continuous observation of the network and (2) minimizing unbrokered network connections.

Index Terms—Internet defense, packet capture, streaming graphs, hypersparse matrices

I. INTRODUCTION

Our community-owned cyber space (industry, government, and academia) requires community-based protections involving significant data sharing [1]–[5]. Large-scale network observations with the highest regard for privacy are key [6]–[8]. In the cyber domain, observatories and outposts have been constructed to gather data on Internet traffic and provide



Fig. 1. Network Sensor Placement. Traffic matrix view of the Internet depicting selected network sensors and their notional proximity to various types of network traffic [9]–[15]. The location of a sensor determines the expected distributions of the observations. Figure adapted from [6].

a starting point for exploring community-based approaches [9]–[15] (see Figure 1). Effective deployment of a sensor requires careful sensor placement, focusing, and calibration with significant volumes of observations. This is particularly important for distributed sensors [16].

The data volumes, processing requirements, and privacy concerns of analyzing a significant fraction of the Internet have been prohibitive. The North American Internet generates billions of non-video Internet packets each second [17], [18]. The GraphBLAS standard provides significant performance and compression capabilities which improve the feasibility of analyzing these volumes of data [19]–[31]. Specifically, the GraphBLAS is ideally suited for both constructing and analyzing anonymized hypersparse traffic matrices. Prior GraphBLAS work has demonstrated rates of 200 billion packets per second (pps) on a supercomputer [32], while achieving compressions of 1 bit per packet [15], and enabling the analysis of the largest publicly available historical archives with over 40 trillion packets [33].

GraphBLAS anonymized hypersparse traffic matrices represent one set of design choices for analyzing network traffic. Specifically, the use case requiring some data on all packets (no down-sampling), high performance, high compression, matrix-based analysis, anonymization, and open standards.

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering. Research was also sponsored by the United States Air Force Research Laboratory and the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

There are a wide range of alternative graph/network analysis technologies and many good implementations achieve performance close to the limits of the underlying computing hardware [34]–[44]. Likewise, there are many network analysis tools that focus on providing a rich interface to the full diversity of data found in network traffic [45]–[47]. Each of these technologies has appropriate use cases in the broad field of Internet traffic analysis.

The ability to rapidly process enormous volumes of anonymized network traffic enables detailed analysis, which is dependent upon proper focusing (understanding precisely where the sensor is) and calibration (understanding the expected statistical distributions of data). This work explores novel focusing and calibration procedures on a multibillion packet dataset using high-performance GraphBLAS anonymized hypersparse matrices.

The outline of the rest of the paper is as follows. First, some basic network quantities are defined in terms of traffic matrices. Next the focusing and calibration procedures are described and demonstrated on a gateway dataset revealing the underlying heavy-tail statistical distributions. A simple model of the corresponding probability of detection (p_d) and probability of false alarm (p_{fa}) for these distributions is presented that highlights the criticality of network sensor focusing and calibration. The paper concludes with a summary and discussion of future work.

II. TRAFFIC MATRICES AND NETWORK QUANTITIES

Internet data must be handled with care. The Center for Applied Internet Data Analysis (CAIDA) based at the University of California's San Diego Supercomputer Center has pioneered trusted data sharing best practices that combine anonymizing source and destinations using CryptoPAN [48] with data sharing agreements. These data sharing best practices include the following principles [6]

- Data is made available in curated repositories
- Using standard anonymization methods where needed: hashing, sampling, and/or simulation
- Registration with a repository and demonstration of legitimate research need
- Recipients legally agree to neither repost a corpus nor deanonymize data
- Recipients can publish analysis and data examples necessary to review research
- Recipients agree to cite the repository and provide publications back to the repository
- Repositories can curate enriched products developed by researchers

Network traffic data can be viewed as a traffic matrix where each row is a source and each column is a destination (see Figure 1). A primary benefit of constructing anonymized hypersparse traffic matrices with the GraphBLAS is the efficient computation of a wide range of network quantities via matrix mathematics. Figure 2 illustrates essential quantities found in all streaming dynamic networks. These quantities are all



Fig. 2. Streaming Network Traffic Quantities. Internet traffic streams of N_V valid packets are divided into a variety of quantities for analysis: source packets, source fan-out, unique source-destination pair packets (or links), destination fan-in, and destination packets. Figure adapted from [49].

TABLE I NETWORK QUANTITIES FROM TRAFFIC MATRICES

Formulas for computing network quantities from traffic matrix \mathbf{A}_t at time t in both summation and matrix notation. 1 is a column vector of all 1's, ^T is the transpose operation, and $| |_0$ is the zero-norm that sets each nonzero value of its argument to 1 [50]. These formulas are unaffected by matrix permutations and will work on anonymized data. Table adapted from [15].

Aggregate	Summation	Matrix
Property	Notation	Notation
Valid packets N_V	$\sum_{i} \sum_{j} \mathbf{A}_{t}(i,j)$	$1^{T}\mathbf{A}_{t}1$
Unique links	$\sum_{i}^{J} \sum_{j}^{J} \mathbf{A}_{t}(i,j) _{0}$	$1^{T} \mathbf{A}_t _01$
Link packets from i to j	$\mathbf{A}_t(i,j)$	\mathbf{A}_t
Max link packets (d_{\max})	$\max_{ij} \mathbf{A}_t(i,j)$	$\max(\mathbf{A}_t)$
Unique sources	$\sum_i \sum_j \mathbf{A}_t(i,j) _0$	$1^{T} \mathbf{A}_t 1 _0$
Packets from source i	$\sum_{i}^{j} \mathbf{A}_{t}(i,j)$	$\mathbf{A}_t 1$
Max source packets (d_{\max})	$\max_i \sum_{j=1}^{j} \mathbf{A}_t(i,j)$	$\max(\mathbf{A}_t 1)$
Source fan-out from i	$\sum_{j}^{j} \mathbf{A}_{t}(i,j) _{0}$	$ \mathbf{A}_t _0 1$
Max source fan-out (d_{\max})	$\max_i \sum_j \mathbf{A}_t(i,j) _0$	$\max(\mathbf{A}_t _0 1)$
Unique destinations	$\sum_{j} \sum_{i} \mathbf{A}_{t}(i,j) _{0}$	$ 1^T \mathbf{A}_t _0 1$
Destination packets to j	$\sum_i \mathbf{A}_t(i,j)$	$1^{T} \mathbf{A}_t _0$
Max destination packets (d_{\max})	$\max_j \sum_i \mathbf{A}_t(i, j)$	$\max(1^{T} \mathbf{A}_t _0)$
Destination fan-in to j	$\sum_{i} \mathbf{A}_{t}(i,j) _{0}$	$1^{T} \mathbf{A}_t$
Max destination fan-in (d_{\max})	$\max_j \sum_i \mathbf{A}_t(i,j) _0$	$\max(1^{T} \mathbf{A}_t)$

computable from anonymized traffic matrices created from the source and destinations found in Internet packet headers.

The network quantities depicted in Figure 2 are computable from anonymized origin-destination traffic matrices that are widely used to represent network traffic [51]–[54]. It is common to filter the packets down to a valid set for any particular analysis. Such filters may limit particular sources, destinations, protocols, and time windows. To reduce statistical fluctuations, the streaming data should be partitioned so that for any chosen time window all data sets have the same number of valid packets [55]. At a given time t, N_V consecutive valid packets are aggregated from the traffic into a hypersparse matrix A_t , where $A_t(i, j)$ is the number of valid packets between the source i and destination j. The sum of all the entries in A_t is equal to N_V

$$\sum_{i,j} \mathbf{A}_t(i,j) = N_V$$

Constant packet, variable time samples simplify the statistical analysis of the heavy-tail distributions commonly found in



Fig. 3. **Multi-temporal streaming traffic matrices.** Efficient computation of network quantities on multiple time scales can be achieved by hierarchically aggregating data in different time windows. Figure adapted from [15].

network traffic quantities [49], [56], [57]. All the network quantities depicted in Figure 2 can be readily computed from \mathbf{A}_t using the formulas listed in Table I. Because matrix operations are generally invariant to permutation (reordering of the rows and columns), these quantities can readily be computed from anonymized data. Furthermore, the anonymized data can be analyzed by source and destination subranges (subsets when anonymized) using simple matrix multiplication. For a given subrange represented by an anonymized hypersparse diagonal matrix \mathbf{A}_r , where $\mathbf{A}_r(i, i) = 1$ implies source/destination *i* is in the range, the traffic within the subrange can be computed via: $\mathbf{A}_r \mathbf{A}_t \mathbf{A}_r$. Likewise, for additional privacy guarantees that can be implemented at the edge, the same method can be used to exclude a range of data from the traffic matrix

$\mathbf{A}_t - \mathbf{A}_r \mathbf{A}_t \mathbf{A}_r$

Efficient computation of network quantities on multiple time scales can be achieved by hierarchically aggregating data in different time windows [55]. Figure 3 illustrates a binary aggregation of different streaming traffic matrices. Computing each quantity at each hierarchy level eliminates redundant computations that would be performed if each packet window was computed separately. Hierarchy also ensures that most computations are performed on smaller matrices residing in faster memory. Correlations among the matrices mean that adding two matrices each with N_V entries results in a matrix with fewer than $2N_V$ entries, reducing the relative number of operations as the matrices grow.

One of the important capabilities of the SuiteSparse Graph-BLAS library is direct support of hypersparse matrices where the number of nonzero entries is significantly less than either dimensions of the matrix [58]. If the packet source and destination identifiers are drawn from a large numeric range, such as those used in the Internet protocol, then a hypersparse representation of A_t eliminates the need to keep track of additional indices and can significantly accelerate the computations [59].

III. NETWORK SENSOR FOCUSING & CALIBRATION

The process for rapid construction and analysis of Graph-BLAS hypersparse traffic matrices is described in [30], [31] and briefly summarized here. The first step in the GraphBLAS hypersparse traffic matrix pipeline is to capture a packet,



Fig. 4. **Parallel Performance.** Single-node parallel performance of the hypersparse GraphBLAS construction program running on multiple distinct PCAP files both with and without direct CryptoPAN anonymization.

discard the data payload, and extract the source and destination Internet Protocol (IP) addresses. For the purposes of the current testing, only IPv4 packets are used which are stored as 32 bit unsigned integers. Collections of $N_V = 2^{17}$ consecutive packets can each be anonymized with CryptoPAN directly or using a CryptoPAN generated anonymization table. The resulting anonymized source and destination IPs are then used to construct a $2^{32} \times 2^{32}$ hypersparse GraphBLAS matrix. 64 consecutive hypersparse GraphBLAS matrices are each serialized in compressed sparse rows (CSR) format with ZSTD [60] compression and saved to a UNIX TAR file.

This procedure [30] was applied to a 246 GB packet capture (PCAP) dataset collected at a gateway for several minutes consisting of $2^{31} = 2,147,483,648$ packets. The hypersparse GraphBLAS matrices were constructed by running the aforementioned procedure on a 32 core 2.3 GHz Xeon 6314 server with 128 GB of RAM and a 7.8 TB NVMe storage device. The resulting 256 compressed GraphBLAS TAR files, each encompassing $64 \times 2^{17} = 8,388,608$ packets, had an average file size of 7.4MB corresponding to < 1 byte per packet. These files were further analyzed [31] to compute all the network quantities in Table I for packet windows $N_V = 2^{17}, \ldots, 2^{27}$ over a 4×4 set of sub-ranges resulting in a single 4.7MB file corresponding to < 0.01 bits per packet. Both of these steps highlight the significant compression (100x and 5000x) of these GraphBLAS analysis procedures on a real-world data set. The single-core single-thread GraphBLAS processing time for this dataset (including IO) was $\approx 3x$ faster than realtime for this 6.8 Gigabit/second network gateway. Figure 4 shows the single-node parallel performance of the hypersparse GraphBLAS construction program running on multiple distinct PCAP files both with and without direct CryptoPAN anonymization.

The focusing procedure begins with understanding the expected location of the network sensor with respect to the various sub-ranges. Figure 5 illustrates a typical position



Fig. 5. **Network Sensor Traffic Matrix Ranges.** The traffic matrix for a given sensor can be divided into ranges separating different categories of traffic. If range 1 is internal traffic, range 2 is at the gateway, and range 3 is external traffic, then a gateway sensor would be expected to see primarily range 2 to range 3 and range 3 to range 2 traffic.

for a gateway sensor. The expectation of such a sensor is that it will primarily see traffic outbound and inbound from the rest of the world. Our test data set is processed using 4 subranges corresponding to internal nonroutable $(2^{24} + 2 + 2^{20} + 2^{16} = 17,891,328$ IP addresses), internal bogon $(2^{22}+2^{17}+2^{16}+6\times2^8=4,392,448$ IP addresses), assigned to the gateway (\approx 3.6million IP addresses), and other addresses corresponding to the rest of the Internet (\approx 4.2 billion IP addresses). A focused traffic matrix will have an observed statistical distribution of packets that is consistent with the location of the sensor.

The focusing procedure can be illustrated in Figure 6 by determining the "endianness" of the data, which may not be apparent in extremely large samples. Generally, network traffic is big-endian in flight, but can be either big-endian or little-endian when stored for processing. Incorrect endianness will defocus the data and make it look more like a random distribution. The big-endian and little-endian representations of the observed data indicate that big-endian is less random and little-endian is more random, thus the underlying data is big-endian given that the sensor is expected to be at a gateway that sees mostly assigned to/from other traffic. The above focusing procedure can be used to correct/check for any distortions in the data and is repeated until the expected distribution is consistent with the known location of the sensor. Likewise, the procedure can be used to quickly determine if a sensor is in a location different from what is expected, which can easily happen when a large number of network sensors are deployed. Once a sensor is focused, the appearance of traffic in sub-ranges where none is expected can be a simple way to detect anomalies. For example, an emerging best-practice is to limit unbrokered network connections [61], which can imply that all traffic must pass through a defined gateway and no other traffic should be observed.

Once the network sensor has been focused, the data in



Fig. 6. **Traffic Matrix Focus Table.** A focused traffic matrix will have an observed statistical distribution of packets that is consistent with the location of the sensor. This can be illustrated with determining the "endianness" of the data. It is possible to estimate the expected distribution of these packets if they are randomly distributed (bottom). Incorrect endianness will defocus the data and make it look more like a random distribution. The big-endian (top) and little-endian (middle) representations of the observed data indicate that big-endian is less random and little-endian is more random, thus the underlying data is big-endian given that the sensor is expected to be at a gateway that sees mostly assigned to/from other traffic.

each of the sub-ranges can be analyzed. Figure 7 shows the distribution of the number of packets going over specific links (source/destination pairs) for different sub-ranges (assigned-to-other and other-to-assigned) and packet window sizes $N_V = 2^{17}, 2^{27}$. These results show the widely observed heavy-tail distributions that are well modeled by the Zipf-Mandelbrot distribution [49], [57]

$$p(d) \propto 1/(d+\delta)^{\alpha}$$

where d is the number packets observed traversing a link and δ and α are best-fit parameters. Figure 7 highlights the strong dependence on the statistical distributions on the sub-range and the packet window size. Given that effective anomaly detection depends upon accurate models of the expected distribution it is worth considering the detection theory implications of these observations.

IV. HEAVY-TAIL DETECTION THEORY

Heavy-tail distributions are widely observed [56], but only recently have these distributions become sufficiently precise that they can be used as background models for detection. Heavy-tail distributions (like Zipf-Mandelbrot) differ significantly from their more commonly known light-tail counterparts (e.g., Gaussian, Poisson, ...), and have many surprising properties: high-likelihood of extreme events, divergent higher-order moments (e.g., infinite variance), and no central limit theorem. In order to understand the implications of heavy-tail distributions on detection, a simple model is presented of the corresponding probability of detection (p_d) and probability of false alarm (p_{fa}) for these distributions.



Fig. 7. **Observed Heavy-Tail Distributions.** Fraction of observed links that have a specified number of packets (d) showing the strong dependence on the size of the packet window (N_V) and direction of the traffic: assigned-to-other (\rightarrow) and other-to-assigned (\leftarrow). Solid black lines show best fit Zipf-Mandelbrot distributions as a function of d.



Fig. 8. Heavy-Tail and Light-Tail Background Model. Heavy-tail distribution (solid blue line) given by a Zipf-Mandelbrot distribution $p_{\rm zm}(d)$ with offset $\delta = 0$ and exponent $\alpha = 2$. Light-tail distribution (solid black line) given by a Gaussian distribution $p_{\rm gauss}(d)$ with mean $\mu = 1$ and variance $\sigma = 0.43$. Dashed lines show upper and lower error bounds on the model as determined by the parameter $c_{\rm err} = 2/3$.

Figure 8 shows representative heavy-tail and light-tail distributions that are each best-fits to underlying heavy-tail observations of the type observed in Figure 7 (clearly the light-tail distribution would be a poor fit). The simple detection theory model begins by specifying upper and lower bounds of the models within which all background and target observations are expected to be found. These ranges are specified by the single parameter $c_{\rm err}$. The bounds are defined in terms of multiplicative factors to represent equal logarithmic spacing which is more consistent with the high dynamic range observed in heavy-tail distributions. Within this range a simple triangular model is used for the expected distribution of background and target observations (see Appendix A). These models can be used with a single threshold parameter $c_{\rm cut}$ to compute the



Fig. 9. Model ROC Curve. Baseline model ($c_{\rm err} = 2/3$, $c_{\rm cut} = 1/3$, $d_{\rm max} = 2^{20}$) illustrating the performance of correct heavy-tail statistics applied to correctly split heavy-tail data. The performance can be further improved by requiring to detections appear in $N_{\rm samp} = 8$ consecutive samples. The performance is significantly degraded if a heavy-tail statistics is applied to mixed (unsplit) such that the underlying heavy-tail model is correct for only half of the bins. The performance is very poor when light-tail statistics are used on heavy-tail data where it is likely only a single bin $f_{\rm d_{max}} = 0.05$ has the correct model.



Fig. 10. Notional ROC Curve. A perhaps more accurate, but less precise version of Figure 9 highlighting that using the correct model on correctly spit data provides good results (heavy-tail statistics with properly split heavy-tail data), using a partially correct model degrades the performance (heavy-tail statistics with mixed heavy-tail data), and that using an entirely incorrect model results in very poor performance (light-tail statistics with heavy-tail data).

expected probability of detection $p_{\rm d}(c_{\rm cut})$ and probability of false alarm $p_{\rm fa}(c_{\rm cut})$ to produce the standard receiver operating characteristic (ROC) curves [62], [63] (see Appendix B).

A simple baseline model covers the case when the background model and the observations are in agreement (see Figure 9). Given the frequency of heavy-tail observations (see Figure 7) and that light-tail statistics are the more common analysis tool, it is important to explore the implications of using an incorrect model on the ROC curve. Assuming the heavy-tail observations have a maximum value of d_{max} and are logarithmically binned into $\log_2(d_{\text{max}})$ bins, a simple model of the effect of using a light-tail distribution on the heavy-tail data is to assume the light-tail distribution is correct for a single bin and incorrect for all others (Figure 8). Effectively, this means that the light-tail model will have a fractional accuracy of $f_{d_{\text{max}}} = 1/\log_2(d_{\text{max}})$. For a typical value of $d_{\text{max}} \approx 2^{20}$, this implies a fractional accuracy of $f_{d_{\text{max}}} \approx 0.05$, which places the ROC curve in either the "all detections" or the "no detections" regime. The corresponding ROC curves for the all detections and no detections cases are shown in Figure 9.

Figure 7 also highlights the importance of correctly sampling and splitting the data to compare with the correct heavy-tail distribution. If the data is not sampled and split correctly, a heavy tail distribution may be observed with significantly different parameters than the appropriate background model. A simple model for this effect can be extrapolated from the previous analysis. Suppose using the incorrect heavy-tail model mischaracterizes half the bins, this is the same as setting $f_{d_{max}} \approx 0.5$ in the previous analysis, whose corresponding ROC curve is also shown in Figure 9.

These simple models provide precise estimates of ROC curves, but the real-world is far more complicated. A notional representation of these ROC curves, which is perhaps more accurate, but less precise is shown in Figure 10. The overall interpretation of Figure 10 is that using the correct model on the correctly split data provides good results (heavy-tail statistics with properly split heavy-tail data), using a partially correct model degrades the performance (heavy-tail statistics with mixed heavy-tail data), and that using an entirely incorrect model results in very poor performance (light-tail statistics with heavy-tail data).

V. CONCLUSIONS AND FUTURE WORK

Community-owned cyber space requires community-based efforts to defend. A key to protecting our shared cyberspace are large-scale network observations that uphold the highest regard for privacy. Careful sensor placement, focusing, and calibration with significant volumes of data are necessary for the deployment network sensors. Novel focusing and calibration procedures using high-performance GraphBLAS anonymized hypersparse matrices are demonstrated on a multibillion packet dataset. Observed real-time processing rates are consistent with high-bandwidth links and show significant data compression, confirming previously reported measurements. The effectiveness of these procedures at focusing the traffic matrix is shown. The underlying stable heavy-tail statistical distributions are also confirmed. Effective anomaly detection depends upon accurate models of the expected distribution and the detection theory implications of these observations are explored with a simple model of the corresponding probability of detection (p_d) and probability of false alarm (p_{fa}) . This model highlights the criticality of network sensor focusing and calibration. In practical terms, once a sensor is properly focused and calibrated it can continuously observe the network and minimize unbrokered network connections, which are two of the central tenets of good cybersecurity.

ACKNOWLEDGMENTS

The authors wish to acknowledge the following individuals for their contributions and support: Daniel Andersen, Sean Atkins, Chris Birardi, Bob Bond, Andy Bowne, Stephen Buckley, K Claffy, Cary Conrad, Chris Demchak, Alan Edelman, Garry Floyd, Jeff Gottschalk, Dhruv Gupta, Chris Hill, Kurt Keville, Charles Leiserson, Kirsten Malvey, Chase Milner, Sanjeev Mohindra, Dave Martinez, Joseph McDonald, Heidi Perry, Christian Prothmann, Steve Rejto, Josh Rountree, Daniela Rus, Mark Sherman, Scott Weed, Adam Wierman, Marc Zissman.

REFERENCES

- S. Atkins and C. Lawson, "An improvised patchwork: success and failure in cybersecurity policy for critical infrastructure," *Public Administration Review*, vol. 81, no. 5, pp. 847–861, 2021.
- [2] S. Atkins and C. Lawson, "Cooperation amidst competition: cybersecurity partnership in the us financial services sector," *Journal of Cybersecurity*, vol. 7, no. 1, 2021.
- [3] C. Demchak, "Achieving systemic resilience in a great systems conflict era," *The Cyber Defense Review*, vol. 6, no. 2, pp. 51–70, 2021.
- [4] S. Weed, "Beyond zero trust: Reclaiming blue cyberspace," Master's thesis, United States Army War College, 2022.
- [5] S. Atkins and C. Lawson, "Beyond zero trust: Reclaiming blue cyberspace with ai," *Cyber Defense Review*, vol. 7, no. 1, 2023.
- [6] J. Kepner, J. Bernays, S. Buckley, K. Cho, C. Conrad, L. Daigle, K. Erhardt, V. Gadepally, B. Greene, M. Jones, R. Knake, B. Maggs, P. Michaleas, C. Meiners, A. Morris, A. Pentland, S. Pisharody, S. Powazek, A. Prout, P. Reiner, K. Suzuki, K. Takhashi, T. Tauber, L. Walker, and D. Stetson, "Zero botnets: An observe-pursue-counter approach." Belfer Center Reports, 6 2021.
- [7] S. Pisharody, J. Bernays, V. Gadepally, M. Jones, J. Kepner, C. Meiners, P. Michaleas, A. Tse, and D. Stetson, "Realizing forward defense in the cyber domain," in 2021 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–7, IEEE, 2021.
- [8] A. Pentland, "Building a new economy: data, ai, and web3," Communications of the ACM, vol. 65, no. 12, pp. 27–29, 2022.
- [9] "CAIDA Anonymized Internet Traces Dataset (April 2008 January 2019)." https://www.caida.org/catalog/datasets/passive_dataset/.
- [10] "UCSD Network Telescope." https://www.caida.org/projects/network_telescope/.
- [11] "Global Cyber Alliance." https://www.globalcyberalliance.org/.
- [12] "Greynoise." https://greynoise.io/.
- [13] "MAWI Working Group Traffic Archive." http://mawi.wide.ad.jp/mawi/.
- [14] "Shadowserver Foundation." https://www.shadowserver.org/.
- [15] J. Kepner, C. Meiners, C. Byun, S. McGuire, T. Davis, W. Arcand, J. Bernays, D. Bestor, W. Bergeron, V. Gadepally, R. Harnasch, M. Hubbell, M. Houle, M. Jones, A. Kirby, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Reuther, A. Rosa, S. Samsi, D. Stetson, A. Tse, C. Yee, and P. Michaleas, "Multi-temporal analysis and scaling relations of 100,000,000,000 network packets," in 2020 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–6, 2020.
- [16] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A collaborative approach to in-place sensor calibration," in *Information Processing* in Sensor Networks: Second International Workshop, IPSN 2003, Palo Alto, CA, USA, April 22–23, 2003 Proceedings, pp. 301–316, Springer, 2003.
- [17] "Cisco Visual Networking Index: Forecast and Trends." https://newsroom.cisco.com/press-release-content?articleId=1955935.
- [18] "Cisco Visual Networking Index: Forecast and Trends, 2018–2023." https://www.cisco.com/c/en/us/solutions/collateral/executiveperspectives/annual-internet-report/white-paper-c11-741490.html.

- [19] J. Kepner, P. Aaltonen, D. Bader, A. Buluç, F. Franchetti, J. Gilbert, D. Hutchison, M. Kumar, A. Lumsdaine, H. Meyerhenke, S. McMillan, C. Yang, J. D. Owens, M. Zalewski, T. Mattson, and J. Moreira, "Mathematical foundations of the graphblas," in 2016 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–9, 2016.
- [20] A. Buluç, T. Mattson, S. McMillan, J. Moreira, and C. Yang, "Design of the graphblas api for c," in 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 643–652, 2017.
- [21] C. Yang, A. Buluç, and J. D. Owens, "Implementing push-pull efficiently in graphblas," in *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1–11, 2018.
- [22] J. Kepner and H. Jananthan, Mathematics of big data: Spreadsheets, databases, matrices, and graphs. MIT Press, 2018.
- [23] T. A. Davis, "Algorithm 1000: Suitesparse: Graphblas: Graph algorithms in the language of sparse linear algebra," ACM Transactions on Mathematical Software (TOMS), vol. 45, no. 4, pp. 1–25, 2019.
- [24] T. Mattson, T. A. Davis, M. Kumar, A. Buluc, S. McMillan, J. Moreira, and C. Yang, "Lagraph: A community effort to collect graph algorithms built on top of the graphblas," in 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 276–284, IEEE, 2019.
- [25] P. Cailliau, T. Davis, V. Gadepally, J. Kepner, R. Lipman, J. Lovitz, and K. Ouaknine, "Redisgraph graphblas enabled graph database," in 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 285–286, IEEE, 2019.
- [26] T. A. Davis, M. Aznaveh, and S. Kolodziej, "Write quick, run fast: Sparse deep neural network in 20 minutes of development time via suitesparse: Graphblas," in 2019 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–6, IEEE, 2019.
- [27] M. Aznaveh, J. Chen, T. A. Davis, B. Hegyi, S. P. Kolodziej, T. G. Mattson, and G. Szárnyas, "Parallel graphblas with openmp," in 2020 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing, pp. 138–148, SIAM, 2020.
- [28] B. Brock, A. Buluç, T. G. Mattson, S. McMillan, and J. E. Moreira, "Introduction to graphblas 2.0," in 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 253–262, IEEE, 2021.
- [29] M. Pelletier, W. Kimmerer, T. A. Davis, and T. G. Mattson, "The graphblas in julia and python: the pagerank and triangle centralities," in 2021 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–7, 2021.
- [30] M. Jones, J. Kepner, D. Andersen, A. Buluç, C. Byun, K. Claffy, T. Davis, W. Arcand, J. Bernays, D. Bestor, W. Bergeron, V. Gadepally, M. Houle, M. Hubbell, H. Jananthan, A. Klein, C. Meiners, L. Milechin, J. Mullen, S. Pisharody, A. Prout, A. Reuther, A. Rosa, S. Samsi, J. Sreekanth, D. Stetson, C. Yee, and P. Michaleas, "Graphblas on the edge: Anonymized high performance streaming of network traffic," in 2022 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–8, 2022.
- [31] T. Trigg, C. Meiners, S. Pisharody, H. Jananthan, M. Jones, A. Michaleas, T. Davis, E. Welch, W. Arcand, D. Bestor, W. Bergeron, C. Byun, V. Gadepally, M. Houle, M. Hubbell, A. Klein, P. Michaleas, L. Milechin, J. Mullen, A. Prout, A. Reuther, A. Rosa, S. Samsi, D. Stetson, C. Yee, and J. Kepner, "Hypersparse network flow analysis of packets with graphblas," in 2022 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–7, 2022.
- [32] J. Kepner, T. Davis, C. Byun, W. Arcand, D. Bestor, W. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Reuther, A. Rosa, S. Samsi, C. Yee, and P. Michaleas, "Vertical, temporal, and horizontal scaling of hierarchical hypersparse graphblas matrices," in 2021 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–6, 2021.
- [33] J. Kepner, M. Jones, D. Andersen, A. Buluç, C. Byun, K. Claffy, T. Davis, W. Arcand, J. Bernays, D. Bestor, W. Bergeron, V. Gadepally, M. Houle, M. Hubbell, A. Klein, C. Meiners, L. Milechin, J. Mullen, S. Pisharody, A. Prout, A. Reuther, A. Rosa, S. Samsi, D. Stetson, A. Tse, C. Yee, and P. Michaleas, "Spatial temporal analysis of 40,000,000,000,000 internet darkspace packets," in 2021 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–8, 2021.
- [34] A. Tumeo, O. Villa, and D. Sciuto, "Efficient pattern matching on gpus for intrusion detection systems," in *Proceedings of the 7th ACM International Conference on Computing Frontiers*, CF '10, (New York, NY, USA), p. 87–88, Association for Computing Machinery, 2010.

- [35] M. Kumar, W. P. Horn, J. Kepner, J. E. Moreira, and P. Pattnaik, "Ibm power9 and cognitive computing," *IBM Journal of Research and Development*, vol. 62, no. 4/5, pp. 10–1, 2018.
- [36] J. Ezick, T. Henretty, M. Baskaran, R. Lethin, J. Feo, T.-C. Tuan, C. Coley, L. Leonard, R. Agrawal, B. Parsons, and W. Glodek, "Combining tensor decompositions and graph analytics to provide cyber situational awareness at hpc scale," in 2019 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–7, 2019.
- [37] P. Gera, H. Kim, P. Sao, H. Kim, and D. Bader, "Traversing large graphs on gpus with unified memory," *Proceedings of the VLDB Endowment*, vol. 13, no. 7, pp. 1119–1133, 2020.
- [38] A. Azad, M. M. Aznaveh, S. Beamer, M. Blanco, J. Chen, L. D'Alessandro, R. Dathathri, T. Davis, K. Deweese, J. Firoz, H. A. Gabb, G. Gill, B. Hegyi, S. Kolodziej, T. M. Low, A. Lumsdaine, T. Manlaibaatar, T. G. Mattson, S. McMillan, R. Peri, K. Pingali, U. Sridhar, G. Szarnyas, Y. Zhang, and Y. Zhang, "Evaluation of graph analytics frameworks using the gap benchmark suite," in 2020 IEEE International Symposium on Workload Characterization (IISWC), pp. 216–227, 2020.
- [39] Z. Du, O. A. Rodriguez, J. Patchett, and D. A. Bader, "Interactive graph stream analytics in arkouda," *Algorithms*, vol. 14, no. 8, p. 221, 2021.
- [40] S. Acer, A. Azad, E. G. Boman, A. Buluç, K. D. Devine, S. Ferdous, N. Gawande, S. Ghosh, M. Halappanavar, A. Kalyanaraman, A. Khan, M. Minutoli, A. Pothen, S. Rajamanickam, O. Selvitopi, N. R. Tallent, and A. Tumeo, "Exagraph: Graph and combinatorial methods for enabling exascale applications," *The International Journal of High Performance Computing Applications*, vol. 35, no. 6, pp. 553–571, 2021.
- [41] M. P. Blanco, S. McMillan, and T. M. Low, "Delayed asynchronous iterative graph algorithms," in 2021 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–7, IEEE, 2021.
- [42] N. K. Ahmed, N. Duffield, and R. A. Rossi, "Online sampling of temporal networks," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 15, no. 4, pp. 1–27, 2021.
- [43] A. Azad, O. Selvitopi, M. T. Hussain, J. R. Gilbert, and A. Buluç, "Combinatorial blas 2.0: Scaling combinatorial algorithms on distributedmemory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 989–1001, 2021.
- [44] D. Koutra, "The power of summarization in graph mining and learning: smaller data, faster methods, more interpretability," *Proceedings of the VLDB Endowment*, vol. 14, no. 13, pp. 3416–3416, 2021.
- [45] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and ipfix," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.
- [46] R. Sommer, "Bro: An open source network intrusion detection system," Security, E-learning, E-Services, 17. DFN-Arbeitstagung über Kommunikationsnetze, 2003.
- [47] P. Lucente, "pmacct: steps forward interface counters," Tech. Rep., 2008.
- [48] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks*, vol. 46, no. 2, pp. 253–272, 2004.
- [49] J. Kepner, K. Cho, K. Claffy, V. Gadepally, P. Michaleas, and L. Milechin, "Hypersparse neural network analysis of large-scale internet traffic," in 2019 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–11, 2019.
- [50] J. Karvanen and A. Cichocki, "Measuring sparseness of noisy signals," in 4th International Symposium on Independent Component Analysis and Blind Signal Separation, pp. 125–130, 2003.
- [51] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, "How to identify and estimate the largest traffic matrix elements in a dynamic environment," in ACM SIGMETRICS Performance Evaluation Review, vol. 32, pp. 73–84, ACM, 2004.
- [52] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating pointto-point and point-to-multipoint traffic matrices: an information-theoretic approach," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 5, pp. 947–960, 2005.
- [53] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *science*, vol. 328, no. 5980, pp. 876–878, 2010.
- [54] P. Tune, M. Roughan, H. Haddadi, and O. Bonaventure, "Internet traffic matrices: A primer," *Recent Advances in Networking*, vol. 1, pp. 1–56, 2013.

- [55] J. Kepner, V. Gadepally, L. Milechin, S. Samsi, W. Arcand, D. Bestor, W. Bergeron, C. Byun, M. Hubbell, M. Houle, M. Jones, A. Klein, P. Michaleas, J. Mullen, A. Prout, A. Rosa, C. Yee, and A. Reuther, "Streaming 1.9 billion hypersparse network updates per second with d4m," in 2019 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–6, 2019.
- [56] J. Nair, A. Wierman, and B. Zwart, "The fundamentals of heavy tails: Properties, emergence, and estimation," *Preprint, California Institute of Technology*, 2020.
- [57] J. Kepner, K. Cho, K. Claffy, V. Gadepally, S. McGuire, L. Milechin, W. Arcand, D. Bestor, W. Bergeron, C. Byun, M. Hubbell, M. Houle, M. Jones, A. Prout, A. Reuther, A. Rosa, S. Samsi, C. Yee, and P. Michaleas, "New phenomena in large-scale internet traffic," in *Massive Graph Analytics* (D. Bader, ed.), pp. 1–53, Chapman and Hall/CRC, 2022.
- [58] A. Buluç, J. T. Fineman, M. Frigo, J. R. Gilbert, and C. E. Leiserson, "Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks," in *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, pp. 233–244, 2009.
- [59] J. Kepner, T. Davis, C. Byun, W. Arcand, D. Bestor, W. Bergeron, V. Gadepally, M. Hubbell, M. Houle, M. Jones, A. Klein, P. Michaleas, L. Milechin, J. Mullen, A. Prout, A. Rosa, S. Samsi, C. Yee, and A. Reuther, "75,000,000,000 streaming inserts/second using hierarchical hypersparse graphblas matrices," in 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 207– 210, 2020.
- [60] Y. Collet and M. Kucherawy, "Zstandard compression and the application/zstd media type," tech. rep., 2018.
- [61] B. Arsenault, "Hybrid workforce security," *Microsoft Digital Defense Report*, pp. 89–108, 2021.
- [62] W. Peterson, T. Birdsall, and W. Fox, "The theory of signal detectability," *Transactions of the IRE professional group on information theory*, vol. 4, no. 4, pp. 171–212, 1954.
- [63] J. Kerekes, "Receiver operating characteristic curve confidence intervals and regions," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 251–255, 2008.

APPENDIX A: BACKGROUND AND TARGET DISTRIBUTIONS

Simple triangular models of the expected distributions around the background model can be created with just two parameters: $c_{\rm err}$ and $c_{\rm cut}$. Let x be observed distance from the expected background distribution p(d), where d is a measured network quantity (see Table I). $c_{\rm err}$ sets the minimum and maximum values of the domains of these distributions.

$$\begin{array}{rcl} 0 < & c_{\rm err} & < 1 \\ x_{\rm min} & = & (1 - c_{\rm err}) \\ x_{\rm max} & = & 1/(1 - c_{\rm err}) \end{array}$$

where $x_{\min} < x < x_{\max}$. c_{cut} sets the minimum and maximum cut values used within the cumulative distributions to label background and targets

$$\begin{array}{rcl} 0 < & c_{\rm cut} & < c_{\rm err} \\ x_{\rm min}^{\rm cut} & = & (1 - c_{\rm cut}) > x_{\rm min} \\ x_{\rm max}^{\rm cut} & = & 1/(1 - c_{\rm cut}) < x_{\rm max} \end{array}$$

For a given value of $c_{\rm cut}$, observations where $x_{\rm min}^{\rm cut} < x < x_{\rm max}^{\rm cut}$ are declared as background and observations where $x < x_{\rm min}^{\rm cut}$ or $x > x_{\rm max}^{\rm cut}$ are declared as targets. From these definitions, the relative, normalized, and cumulative lower/higher background and target models can be computed.



Fig. 11. Relative Background and Target Probabilities. The expected distribution of background and targets as a function of the distance x from the background model p(d) for $c_{\rm err} = 2/3$. $p_{\rm back/tar}^{low/high}(x)$ is the probability of a background/target being lower/higher that the model. Background is expected to peak around the background model while targets are more likely away from the background model.

(1) Relative, normalized, cumulative lower background model

$$p_{\text{back}}^{\text{low}}(x) \propto (x - x_{\min})/c_{\text{err}} = 1 + (x - 1)/c_{\text{err}}$$

$$p_{\text{back}}^{\text{low}}(x) = 2(x - (1 - c_{\text{err}}))/c_{\text{err}}^2$$

$$P_{\text{back}}^{\text{low}}(x) = (1 - x)(2c_{\text{err}} + x - 1)/c_{\text{err}}^2$$

(2) Relative, normalized, cumulative higher background model

$$\begin{aligned} p_{\rm back}^{\rm high}(x) &\propto 1 - (x-1)/(x_{\rm max}-1) = x + (1-x)/c_{\rm err} \\ p_{\rm back}^{\rm high}(x) &= (2/c_{\rm err})(1-c_{\rm err})(x+(1-x)/c_{\rm err}) \\ P_{\rm back}^{\rm high}(x) &= (1-c_{\rm err})(x-1)(c_{\rm err}x+c_{\rm err}-x+1)/c_{\rm err}^2 \end{aligned}$$

(3) Relative, normalized, cumulative lower target model

$$p_{\text{tar}}^{\text{low}}(x) \propto 1 - (x - x_{\min})/c_{\text{err}} = (1 - x)/c_{\text{err}}$$

$$p_{\text{tar}}^{\text{low}}(x) = 2(1 - x)/c_{\text{err}}^{2}$$

$$P_{\text{tar}}^{\text{low}}(x) = (x - 1)^{2}/c_{\text{err}}^{2}$$

(4) Relative, normalized, cumulative higher target model

$$p_{\text{tar}}^{\text{high}}(x) \propto (x-1)/(x_{\text{max}}-1) = (1-c_{\text{err}})(x-1)/c_{\text{err}}$$

$$p_{\text{tar}}^{\text{high}}(x) = (2/c_{\text{err}}^2)(1-c_{\text{err}})^2(x-1)$$

$$P_{\text{tar}}^{\text{high}}(x) = (c_{\text{err}}-1)^2(x-1)^2/c_{\text{err}}^2$$

Figure 11 and Figure 12 shows the above distributions for the specific values of $c_{\rm err} = 2/3$ and $c_{\rm cut} = 1/3$.

APPENDIX B: BACKGROUND AND TARGET LABELING

The four outcomes of labeling targets and background can be computed from the corresponding cumulative probabilities $P_{\text{back/tar}}^{low/high}(x > < x_{\min/\max}^{\text{cut}})$. The definitions of the outcomes and their corresponding formula are [63]

TT True target labeled as a target (correct detection)

$$p(\mathbf{TT}) = (1 - P_{\text{tar}}^{\text{low}}(x > x_{\min}^{\text{cut}}) + 1 - P_{\text{tar}}^{\text{high}}(x < x_{\max}^{\text{cut}}))/2$$



Fig. 12. Cumulative Background and Target Probabilities. The expected probability of background and targets relative to x^{cut} threshold for $c_{\text{err}} = 2/3$ and $c_{\text{cut}} = 1/3$. $P_{\text{back/tar}}^{low/high}(x > / < x_{\min/\max}^{\text{cut}})$ is the probability of a background/target being lower/higher than the corresponding cut threshold $x_{\min/\max}^{\text{cut}}$. From these probabilities the four outcomes of labeling can be computed: true targets labeled as targets (TT), true targets labeled as background (TB), true background labeled as target (BT), and true background labeled as background (**BB**).

TB True target labeled as background (missed detection)

$$p(\mathsf{TB}) = (P_{\text{tar}}^{\text{low}}(x > x_{\min}^{\text{cut}}) + P_{\text{tar}}^{\text{high}}(x < x_{\max}^{\text{cut}}))/2$$

BT True background labeled as target (false alarm)

$$p(\mathsf{BT}) = (1 - P_{\text{back}}^{\text{low}}(x > x_{\min}^{\text{cut}}) + 1 - P_{\text{back}}^{\text{high}}(x < x_{\max}^{\text{cut}}))/2$$

BB True background labeled as background (correct nondetection)

$$p(\mathsf{BB}) = (P_{\mathrm{back}}^{\mathrm{low}}(x > x_{\min}^{\mathrm{cut}}) + P_{\mathrm{back}}^{\mathrm{high}}(x < x_{\max}^{\mathrm{cut}}))/2$$

From these definitions, baseline formulas for the probability of detection (p_{det}) and probability of false alarm (p_{fa}) can be derived that are ultimately a function of the single parameter c_{cut} via $x_{min/max}^{cut}$

$$p_{\text{det}}(c_{\text{cut}}) = \frac{N(\mathbf{TT})}{N(\mathbf{TT}) + N(\mathbf{TB})} = p(\mathbf{TT})$$
$$p_{\text{fa}}(c_{\text{cut}}) = \frac{N(\mathbf{BT})}{N(\mathbf{BT}) + N(\mathbf{BB})} = p(\mathbf{BT})$$

where N() is the count of the corresponding outcome. Plotting $p_{det}(c_{cut})$ versus $p_{fa}(c_{cut})$ for different values of c_{cut} produces the standard receiver operating characteristic (ROC) curve [62] (see Figure 9).

A standard enhancement to this model is to integrate over time and assert that true detections are coherent over N_{samp} consecutive samples. This does not change p_{det} but can have a significant effect on p_{fa} . Assuming the background is random and observations are binomially distributed between **BB** and **BT**, the probability that a true background will be randomly categorized as a detection in k out of N_{samp} is given by

$$p(n(\mathbf{BB}) = k) = \binom{N_{\text{samp}}}{k} p(\mathbf{BB})^k (1 - p(\mathbf{BB})^{N_{\text{samp}} - k}$$

If an observation must appear as a detection in all N_{samp} to be a true detection, then a false alarm requires k = 0 resulting in

$$p_{fa}(c_{cut}) = p(n(\mathbf{BB}) = 0)$$

= $(1 - p(\mathbf{BB}))^{N_{samp}}$
= $p(\mathbf{BT})^{N_{samp}}$

since p(BT) = 1 - p(BB). This has a significant positive impact on the shape of the ROC curve (see Figure 9).

The baseline model covers the case when the background model and the observations are in agreement. Given the frequency of heavy-tail observations (see Figure 7) and that light-tail statistics are the more common analysis tool, it is important to explore the ROC curve implications of using a light-tail model on heavy-tail observations. Assuming the heavy-tail observations have a maximum value of d_{max} and are logarithmically binned into $\log_2(d_{\text{max}})$ bins, a simple model of the effect of using a light-tail distribution on the heavytail data is to assume the light-tail distribution is correct for a single bin and incorrect for all others (Figure 8). Effectively, this means that the light-tail model will have a fractional accuracy of $f_{d_{\text{max}}} = 1/\log_2(d_{\text{max}})$. For a typical value of $d_{\rm max} \approx 2^{20}$, this implies a fractional accuracy of ${\rm f}_{d_{\rm max}} \approx 0.05$, which places the ROC curve in either the "all detections" or the "no detections" regime. The all detections regime can be estimated by declaring all observations not in the correctly modeled bin as detections because they satisfy the $x < x_{\min}^{\text{cut}}$ or $x > x_{\max}^{\text{cut}}$ criteria. Only $f_{d_{\max}}$ of the observations could possibly be categorized with the model and the remaining $1 - f_{d_{\max}}$ will be declared detections

$$p_{\text{det}}(c_{\text{cut}}) = p(\mathbf{TT}) \mathbf{f}_{d_{\text{max}}} + 1 - \mathbf{f}_{d_{\text{max}}}$$
$$p_{\text{fa}}(c_{\text{cut}}) = p(\mathbf{BT}) \mathbf{f}_{d_{\text{max}}} + 1 - \mathbf{f}_{d_{\text{max}}}$$

Likewise, the no detections regime can be estimated by declaring all observations not in the correct bin as non-detections because they lie outside the domain of the model the $x_{\min} < x < x_{\max}$. Only $f_{d_{\max}}$ of the observations could possibly be categorized with the model and the remaining $1 - f_{d_{\max}}$ will be declared non-detections

The corresponding ROC curves for the all detections and no detections cases are shown in Figure 9.