Vulnerability Analysis of High Dimensional Complex Systems

Vedant Misra, Dion Harmon, and Yaneer Bar-Yam

New England Complex Systems Institute
{vedant,dion,yaneer}@necsi.edu
http://www.necsi.edu

Abstract. Complex systems experience dramatic changes in behavior and can undergo transitions from functional to dysfunctional states. An unstable system is prone to dysfunctional collective cascades that result from self-reinforcing behaviors within the system. Because many human and technological civilian and military systems today are complex systems, understanding their susceptibility to collective failure is a critical problem. Understanding vulnerability in complex systems requires an approach that characterizes the coupled behaviors at multiple scales of cascading failures. We used neuromorphic methods, which are modeled on the pattern-recognition circuitry of the brain and can find patterns in high-dimensional data at multiple scales, to develop a procedure for identifying the vulnerabilities of complex systems. This procedure was tested on microdynamic Internet2 network data. The result was a generic pipeline for identifying extreme events in high dimensional datasets.

Keywords: complex systems, vulnerability detection, stability and instability, high-dimensional, dimensionality reduction, neuromorphic methods, self-stabilizing systems.

1 Introduction

High dimensional complex systems are comprised of large numbers of interdependent elements [9]. When high dimensional systems perform critical tasks, the task is shared by and dynamically allocated among the components. The ability to distribute function dynamically enables robust and self-stabilizing function in a highly variable environment, but breaks down when collective loads are excessive, or when local failures or allocation process failures lead to cascading failures of large parts of the system as a whole. Thus, interdependence is necessary for function, but at the same time leads to dysfunctions associated with collective breakdowns. Because collective failures are dynamic and emergent, it is essential to identify when they occur and how to prevent them for the effective operation of a large number of critical systems.

Predicting the conditions of collective failures typically requires extensive study of the system and an understanding of both general dynamical characteristics and specific structural details. This is apparent in the limited prediction ability of such well-known collective failures as traffic jams and gridlock in

S. Dolev et al. (Eds.): SSS 2010, LNCS 6366, pp. 560–572, 2010.

road and highway systems. Similar issues arise in many much less visible systems, including power grids; water supply systems; communication networks (the internet); transportation networks (airlines, trains, shipping, etc); the global financial system; manufacturing, food and other commodity supply systems; and social networks and organizations. The potential impact of catastrophic failures in such systems has led to interest in developing detailed models of the systems, but not principles for evaluating system vulnerability [16,37].

Generally, interactions among a system's elements can generate collective dysfunctions, and operating conditions can trigger dramatic changes in the system's overall behavior, such as cascading failures. When a system is highly susceptible to behavioral changes of this sort, it is functionally unstable [20,29].

Vulnerable systems are likely to transition from stability to instability. Like a pencil on its tip, a vulnerable system will collapse if it experiences a sufficiently large deviation. By contrast, a stable system can restore itself to its equilibrium state when perturbed, like a pendulum. A system that is normally stable can become functionally unstable due to changes in global conditions or in relationships between the system's constituent elements.

Understanding the vulnerabilities of complex systems is a critical societal problem because of the many human and technological systems today that rely on distributed function and that can be characterized as high dimensional complex systems. Currently, responses to failure are reactive instead of proactive because we do not have a generic pipeline for analyzing high dimensional systems and anticipating their vulnerabilities. The goal of this paper is to develop a method for characterizing and anticipating extreme behavior and system failure, and to test it on a specific case study.

2 Internet2

Transitions from stability to instability are manifest in the Internet, which makes it a suitable prototype case for studying the dynamical properties of high dimensional systems [11,21,30,32,36]. A central function of the Internet is to enable any node to communicate with any other node transparently and without significant delays or lost communication. The Internet is designed as a self-stabiizing system [6], returning by itself to normal operation despite data errors and equipment failure [31] and despite dynamical deviations from functional states [14]. Nonetheless, the Internet architecture sometimes exhibits collective behaviors that make transparent end-to-end connectivity impossible. Such aggregate collective phenomena include cascading failures [20,22,29], the largest of which have been associated with worm attacks [12,35,38], and "route flapping," which occurs when a router fluctuates quickly between routes without settling into an effective routing pattern [24]. Other such phenomena include bottlenecks, storms, and collective oscillations[10,17,25].

A suitable prototype case for studying the dynamical properties of the Internet is the Internet2 network, backbone hubs of which are depicted in Figure 2. Internet2 is a collaboration of research institutions and companies that require

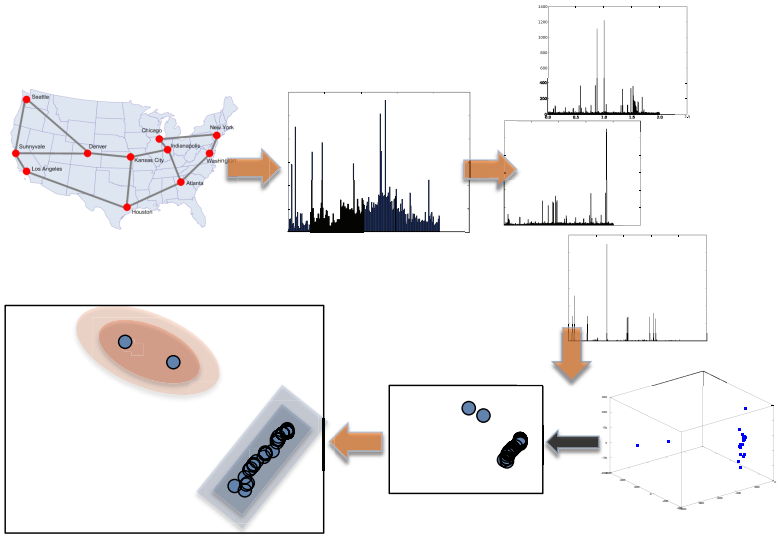

Fig. 1. Depiction of the pipeline. Each orange arrow represents one phase of the four-step process. Step 1, the sensor process, converts high dimensional heterogeneous data into a structured data stream representation. Step 2, the attention process, determines an attention trigger, extracts high-dimensional event data, and applies an alignment algorithm to align events in time. The result is a high-dimensional matrix. Step 3, the pattern process, employs pattern-discovery algorithms (gray arrow) to convert the high dimensional input into a lower-dimensional representation. Step 4, the interpretation process, characterizes the domains in the lower-dimensional representation space and makes it possible to distinguish normal system operations from system vulnerability or failure.

Fig. 2. Backbone hubs on the Internet2 network and traffic flow links between them

high-speed network infrastructure for communication. The Internet2 network is similar to the Internet in design and function, but smaller. While Internet2 is partially isolated from the Internet, it uses the same protocols for routing and is large enough to manifest collective dysfunction [18]. By design, massive volumes of network data can be collected through protocols built into the network, so extensive data about traffic on Internet2 has been archived [1,2,3,4,5,13]. The availability of historical data makes Internet2 a suitable laboratory for studying the collective failure of high-dimensional systems [23,33,39].

Internet2 data archives include logs of routing changes in the network issued by its communications protocol, Border Gateway Protocol (BGP) the same protocol used by routers on the Internet [34]. Under BGP, each node sends updates to its neighbors about which routes are most efficient for transmitting data. As the traffic demand changes, routes can become overloaded. Delays are detected by network routers that read messages from other routers. When delays are detected, BGP messages are sent between routers so that they change their routing tables [34].

Consider what might happen during a speech given by the President that is broadcast via a live video feed from Washington and is of interest to many people near Kansas City. Data packets may be transmitted from Washington, through Atlanta and Houston, to Kansas City (see Fig. 2). The resulting spike in network traffic may impede traffic from Indianapolis to Washington, which also passes through Atlanta.

To overcome this problem, Indianapolis traffic may be rerouted through Chicago and New York. A message from Atlanta to Indianapolis forcing this routing change constitutes a BGP update. The number of updates per minute varies from as little as a few dozen to several thousand depending upon the volume and nature of network activity. System failure occurs if the network experiences unusually high update volumes without settling into an effective routing pattern.

A simple example of system failure is one that occurs in self-generated traffic and route oscillations, where if one route is overloaded, the system dynamically reroutes traffic. However, rerouted traffic may cause overloading and delays in the new route while leaving the older route underutilized. Subsequent rerouting may exacerbate this effect by inducing routing oscillations that never achieve effective system utilization.

Update logs can in principle enable an observer to understand the network's dynamics. However, a single update, or even a large number of updates, are not indicative of failure. Aggregate behaviors must be characterized using patterns in the BGP traffic that enable us to distinguish poor resource utilization and failed communications from effective use of bandwidth and successful communications.

BGP updates are one of several types of records in the Internet2 archives. A central problem in developing a model that reveals the network's collective behaviors is determining which data best represent the system and which can be ignored. Additionally, understanding vulnerabilities in Internet2 requires an approach that recognizes the consequences of dependencies between nodes.

Traditional analysis, which focuses on individual variables and pair-wise correlations, is not sufficient to capture the system's collective behaviors and does little to help discriminate between useful and irrelevant data streams. Furthermore, collective behaviors at multiple scales should be described by k-fold correlations [7,8,15,26,27] that would be difficult to evaluate directly.

3 Neuromorphic Method

We have developed a process for identifying extreme behavior in high-dimensional systems using neuromorphic pattern discovery methods. This process characterizes the differences between patterns of collective behavior and uses them to recognize instability.

Neuromorphic pattern discovery methods are designed to mimic the nervous system's pattern-recognition circuitry using computer algorithms. Our approach consists of four stages: sensor, attention (event detection), pattern finding, and interpretation (classification). Each of these stages is analogous to a specific neurobiological function.

This report describes the successful implementation of our approach but does not describe the multiple methods that have been studied in order to develop this approach [9,28]. These studies investigated both conceptual and practical aspects of computational analysis. Some of the implementations tested in order to identify the strategy used and its refinement were performed on systems other than the Internet 2 data reported here.

Optimization of the method has been performed at a global rather than a local scale, which ensures that the neuromorphic method retains essential information while eliminating unnecessary or redundant information at each stage of processing. That the method does not require optimization at each stage is critical to its widespread applicability. Thus, in this method, no attempt is made at each phase of the process to isolate a single correct output, because a multiplicity of potential outputs can, after the interpretation process has been applied, result in the same conclusion.

We ensure that the patterns discovered by the process are meaningful by requiring that we retain key representative elements of the data stream. High dimensional data is retained until the penultimate stage. Information selection at earlier stages is designed to retain a representation of the coupled dynamical processes that underly system failure. The relative timing of events among multiple units is a critical aspect of the information retained that is often discarded in other forms of analysis. The relative timing data contains the high order correlations among the components of the system.

We overcome the difficulty faced by pattern recognition methods in resolving patterns where multiple instances of the same phenomenon do not appear the same in the input due to transformations such as time or space translation. To address this limitation, we treat the overall collective dynamics of the system as a single entity. We implement a symmetry-breaking process that aligns the events with each other in time. Such a symmetry-breaking process could also have been done in space, but was not necessary for this application.

3.1 Sensor Process

The sensor process refines large volumes of variously structured raw data into a well-defined and standardized high-dimensional parallel data stream. This is analogous to the brain converting compressions and rarefactions of air molecules against the eardrum, or light waves reaching the retina, into neural signals. The biological examples demonstrate that this stage of the process is system specific– i.e. the nature of the originating data is specific to the system being considered (sound or light) and the purpose of the sensory stage is to use a system-specific mechanism to convert the available information into a formatted data stream.

The 10 TB of available data for Internet2 were refined by a computer program that processed raw network data into a dynamic measure of network interactions while dealing with complications like data inconsistencies and gaps. The available data consist of second-resolution logs of various network statistics, including netflow data containing a record of IP flows passing through each router, throughput data consisting of records of the average rate of successful message delivery, and usage data comprised of logs of system load for individual machines at each node. The sensor program parsed these data and returned a time series of the most representative aspects of the data set for the collective behavior with which we are concerned — a data stream representing the existence of a change in the router table at a particular router of the system.

3.2 Attention Process

The next phase of processing requires that we specify a "trigger,"—a dynamic feature of large excursions that we can use to identify when an extreme event may be happening. The trigger is tuned using historical data to maximize the number of events identified by the event detection process while excluding false positives from the data set.

The trigger is based on an aggregate measure of the system's behavior over space and time – in the case of Internet2, across major backbone nodes. Event data is extracted from the data stream using a program that monitors this aggregate measure. A deviation of the measure from a background value well above its statistical variation signals an event – we looked for deviations larger than 3 standard deviations above the moving average – at which point the event's data stream is extracted. Figure 3 is a visualization of an aggregate measure of the behavior of the system, in which each bar represents the number of update messages per day over seven months. The figure shows that update spikes are an easily-identified first approximation for what might constitute an appropriate trigger.

The next phase is to align the event data; this is an essential part of the attention process because it enables comparison of the intra-event dynamics of different events. An algorithm extracts a 40-hour window of data surround each event and examines it to identify the period within that window that best represents aberrant network activity, and then shifts each window in time according to the location of the most active period. In the example in Figure 4, the windows have been shifted to align the largest spikes within the window.

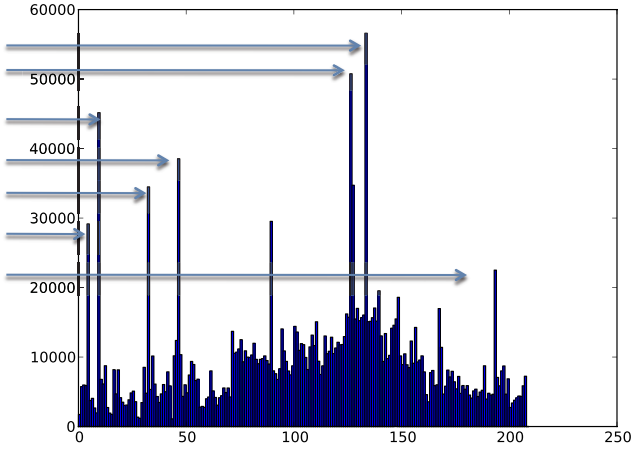

Fig. 3. BGP Updates per day over seven months at an Internet2 node; x-axis is days since the start of 2008. Arrows indicate days that are visually identifiable as "spikes" in the number of BGP updates. The attention process in a neuromorphic algorithm must identify the quantitative signature of such anomalies and use it to extract a data stream that represents the network's dynamical properties.

The alignment process outputs a set of 15,000-element vectors, one vector per event. Each vector represents the behavior at every node over a specific time frame, with the salient features of each event aligned within the output matrix.

3.3 Pattern Process

To identify details of the dynamics of large excursions, we employ a wide array of pattern finding algorithms designed for processing high dimensional, high volume data. Many of these algorithms reduce the dimensionality of the system description by discarding dimensions that are not essential for characterizing the system's overall behavior. A common approach to dimensionality reduction is to assume that the data lie on an embedded non-linear manifold within the high-dimensional space defined by the complete dataset. While some techniques give a mapping from the high dimensional space to the low dimensional space, others provide only a visualization of the low-dimensional data.

Both types of algorithms are designed to maximize coverage of the lower dimensional representation space and minimize the distortion of the projection. Dimensionality reduction algorithms map high-dimensional data vectors ξ_i in an input space of dimension n to lower-dimensional representation vectors x_i in an output space of dimension $m << n$. The algorithms seek to preserve the distances between pairs of points. Given metrics d_ξ and d_x that measure distances between high-dimensional vectors and low-dimensional vectors, respectively, the distances $d_x(x_i, x_j)$ approximate the distances $d_\xi(\xi_i, \xi_j)$. At the same time the algorithms try to maximize a measure of spatial covering so that the

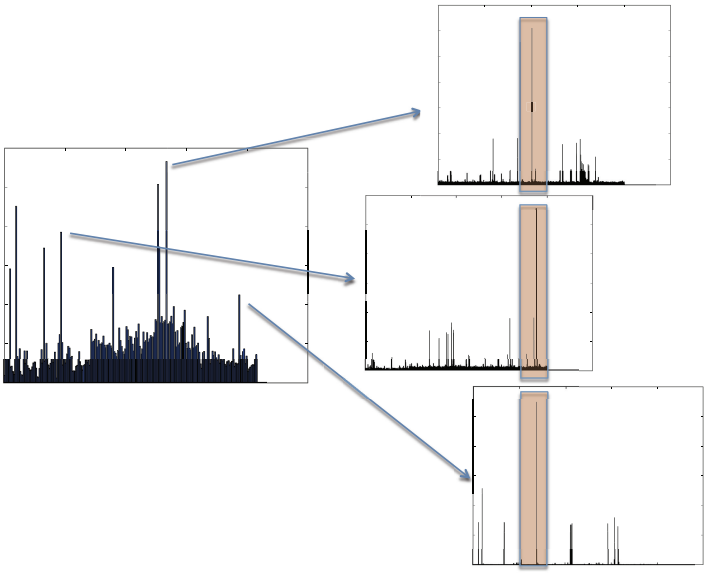

Fig. 4. Aligning events using their dynamic profiles. The window at left depicts the dynamic profile of an Internet2 node over several months. The windows at right depict the profiles of individual days during the month that were flagged during event detection. The alignment process determines what significant characteristic of each event best correlates to significant characteristics of other events and aligns them using the resulting criterion.

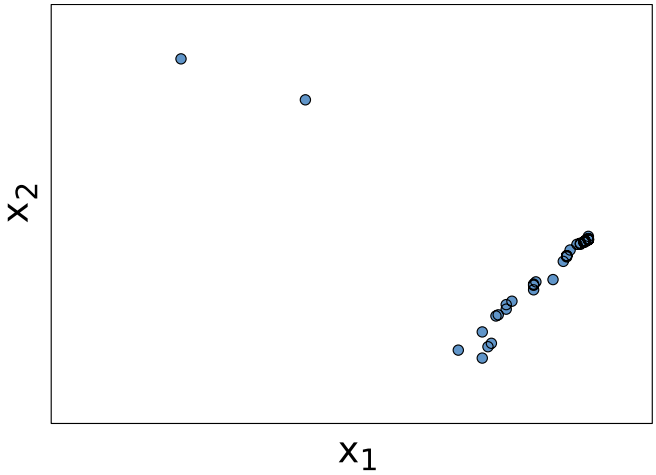

Fig. 5. Dimensionality-reduced representation of event vectors. Each point represents an event. This plot represents an attribute space of event parameters found to be significant by the dimensionality reduction algorithm. Specifically, x_1 and x_2 are the two most prominent lower-dimensional parameters. Note the two extreme events, which appear separate from the large number of rerouting events that did not destabilize the network.

representation vectors x_i represent as much of the high dimension variation in the lower dimension as possible.

One technique for maximizing spatial covering employed by some dimensionality reduction algorithms is to use an intermediary transformation from the input space to a *feature space* in which the underlying structure of the input vectors is more visible. This enables non-linear methods to be incorporated in an otherwise linear process. One such method is Kernel Principal Component Analysis (Kernel PCA), in which the linear operations of PCA are applied to the feature space with nonlinear mapping. Given a set of input data points $\xi_i, i = 1, 2, \ldots, n$ in the n-dimensional input space, we would first nonlinearly transform the i^{th} input vectors ξ_i into a point $\Phi(\xi_i)$ in an N_H dimensional feature space H where each

$$\Phi(\xi_i) = (\phi_1(\xi_i), \ldots, \phi_{N_H}(\xi_i)) \in H, \quad i = 1, 2, \ldots, n. \tag{1}$$

and then use PCA in the feature space H [19]. Carrying out linear PCA in the feature space then yields a presumably lower-dimensional distance-preserving representation of the input vectors x_i, $i = 1, 2, \ldots, m$, with $m < n$. The method we employed was inspired by Kernel PCA; the nonlinear sensor and attention processes primed the input space for dimensionality reduction, after which linear methods were sufficient for identifying structure in the data.

The results of nonlinear dimensionality reduction are visualizations of the high dimensional data in a lower dimensional space that make it possible to uncover patterns within the data using the coordinates of the resulting points in the low dimensional space. Figure 5 depicts a scatter plot generated using the results of dimensionality reduction.

3.4 Interpretation Process

The pattern finding process outputs a representation of the lower-dimensional space to which high-dimensional input was mapped. Just as in a neural processing system, interpration of this lower-dimensional representation must be guided by an understanding of the consequences of previous events, either by studying long term feedback or by training from a previous generation. Similarly, the interpretation of the events in the neuromorphic system can be guided by human interpretation. Since the dimensionality of this output space is small, our own interpretive processes can identify the relevant regions of the space from the historical data.

Extreme events appeared separate from the cluster of background events in the lower-dimensional output space. They are visible in Figure 5 in the upper-left. To determine what property of these events separates them from the trend, we used radar charts that illustrated the node-to-node variation of each event and temporal plots to visualize the dynamics of the activity.

Figure 6 contains two such plots, along with temporal plots and indications of where each event falls in Figure 5. The radar chart insets indicate the magnitude of each event at each node. Clearly visible in the first of the two events, which manifested at every node but was aberrantly large at only one node, are several

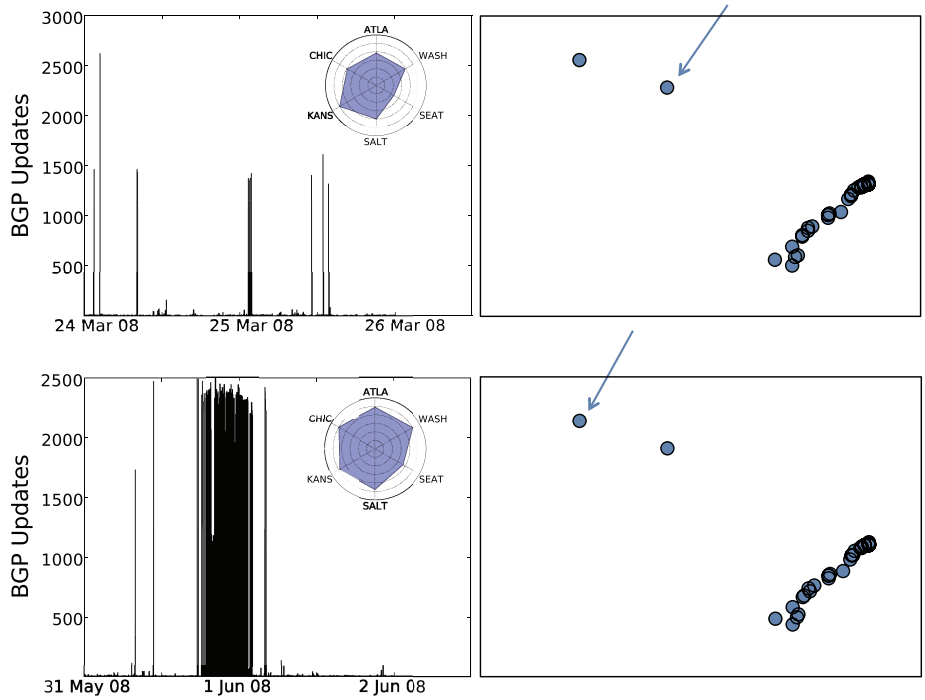

Fig. 6. Dynamic profiles for the two events. The insets indicate the magnitude of each event at each node. Also indicated is where each event appeared on the dimensionality-reduced plot.

distinct spikes in network activity. This repeated and persistent aberrant activity is a signature of systemic instability. The second event depicted consisted of nearly eight hours of large numbers of updates. The entire network was forced to completely rewire itself every 30 seconds. This is precisely the type of system failure our method is designed to detect.

The results of our analysis prompted us to revisit the theoretical nature of vulnerability and failure. Within the context of vulnerable systems, large cascades are common rather than isolated events. System failure is a persistent and recurrent cascade. Thus, both vulnerability and failure can be identified as persistent large deviations from normal behavior.

Our analysis shows that self-stabilizing systems can be vulnerable to collective dysfunctions. While the routing systems can adapt rapidly to changes in the network and the dynamics of demand, there are conditions of the system or the demand on the system that can lead to cascades that cause dysfunction. Recognizing these conditions and detecting extreme events is essential to expanding the domain of effective function.

Our processing pipeline is well-suited to detecting extreme events because of the attentional trigger which aligns events according to their largest excursion.

All events where the large excursion is sufficiently isolated from other large events will appear much more similar to each other than they do to events for which multiple large excursions occur over time and across the network. This ensures that the pattern recognition algorithms will be able to distinguish between the two types.

Our method sheds light on the dynamical characteristics of extreme events and explains why our processing pipeline can distinguish extreme events from those that do not result in system failure. This new insight provides a general explanation of how and why real-time detection of extreme events is possible.

4 Conclusion

We have developed a neuromorphic information processing pipeline that can characterize the vulnerability of complex systems. The process consists of extracting a dynamic measure of network activity and processing the resulting time series to find patterns of collective behavior. The process succeeded in identifying extreme events that are distinct from high demand but otherwise effective system activity. Novel spatiotemporal analysis and dimensionality reduction techniques made this result possible. The pipeline can be used quite generally for analyzing high-dimensional time series and isolating extreme events in real world communication, transportation and economic systems. This system can be combined with real-time system monitoring of data streams to identify dysfunctional behaviors and characterize vulnerabilities or system instabilities as they occur.

Acknowledgments

This work was supported in part by the Defense Advanced Research Projects Agency, the U. S. Army Research Laboratory, and the U. S. Army Research Office under contract/grant number W911NF-09-1-0165.

References

1. B.G.P.: routing table analysis. modified (August 2006), http://thyme.apnic.net/
2. BGPmon: Next generation BGP Monitor, http://bgpmon.netsec.colostate.edu/
3. Border gateway protocol (BGP) data collection standard communities, http://www.bgp4.as/bgp-data-collection-standard-communities (modified February 23, 2006)
4. New sources of BGP data, http://inl.info.ucl.ac.be/blogs/08-10-27-new-sources-bgp-data (modified October 28, 2008)
5. University of Oregon Route Views Project, http://www.routeviews.org/ (modified January 25, 2005)
6. Adam, C., Stadler, R.: Patterns for routing and self-stabilization. In: Proc IEEE/IFPS NOMS (2004)

7. Bar-Yam, Y.: Multiscale complexity/entropy. Advances in Complex Systems 7, 47–63
8. Bar-Yam, Y.: Multiscale variety in complex systems. Complexity 9(4), 37–45 (2004)
9. Bar-Yam, Y.: Dynamics of Complex Systems. Perseus Press, Cambridge (1997)
10. Barabási, A.L., de Menezes, M., Balensiefer, S., Brockman, J.: Hot spots and universality in network dynamics. The European Physical Journal B 38, 169–175 (2004)
11. Cowie, J., Ogielski, A., Premore, B., Yuan, Y.: Global routing instabilities triggered by Code Red II and Nimda worm attacks
12. Cowie, J., Ogielski, A., Premore, B., Yuan, Y.: Internet worms and global routing instabilities: scalability and traffic control in IP networks II. In: Proc SPIE, vol. 4868, pp. 195–199 (2002)
13. Cymru, T.: BGP monitoring, `http://www.team-cymru.org/Monitoring/BGP/` (modified 2010)
14. Dolev, S.: Self-stabilization. MIT Press, Cambridge (2000)
15. Gheorghiu-Svirchevski, S., Bar-Yam, Y.: Multiscale analysis of information correlations in an infinite-range, ferromagnetic ising system. Phys. Rev. E 70(066115) (2004)
16. Hohn, N.: Measuring, understanding, and modelling internet traffic
17. Huberman, B., Lukose, R.: Social dilemmas and internet congestion. Science 277 (1997)
18. Internet2: `http://www.internet2.edu`
19. Izenman, A.: Modern Multivariate Statistical Techniques. Springer, Heidelberg (2008)
20. Jr., E.C., Ge, Z., Misra, V., Towsley, D.: Network resilience: exploring cascading failures within bgp. In: Proceedings of Allerton Conference on Communications, Computing, and Control (2001)
21. Labovitz, C., Malan, G., Jahanian, F.: Internet routing instability 6(5) (1998)
22. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. BUCS (002) (2005)
23. M G.:
24. Mao, Z., Govindan, R., Varghese, G., Katz, R.: Route flap damping exacerbates internet routing convergence. In: Proceedings of ACM SIGCOMM, Pittsburgh, PA, USA, pp. 221–233 (2002)
25. de Menezes, M., Barabási, A.L.: Fluctuations in network dynamics. Phys. Rev. Lett. 92(028701) (2008)
26. Metzler, R., Bar-Yam, Y.: Multiscale analysis of correlated gaussians. Phys. Rev. E 71(046114), 2005 (2005)
27. Metzler, R., Bar-Yam, Y., Kardar, M.: Information flow through a chaotic channel: prediction and postdiction at finite resolution. Phys. Rev. E 70(020605) (2004)
28. Misra, V., Harmon, D., de Aguiar, M., Epstein, I., Braha, D., Bar-Yam, Y.: Vulnerability detection in complex systems. Unpublished report (2009)
29. Motter, A., Lai, Y.C.: Cascade-based attacks on complex networks. Phys. Rev. E 66(065102) (2002)
30. Nicol, D.: Challenges in using simulation to explain global routing instabilities. In: Conference on Grand Challenges in Simulation (2002)
31. Perlman, R.: Interconnections: Bridges, Routers, Switches and Internetworking Protocols, 2nd edn. Addison-Wesley Longman, Amsterdam (2001)
32. Valverde, S., Internet's, R.S.: critical path horizon. The European Physical Journal B 38(2) (2004)

33. Siganos, G., Faloutsos, M.: Detection of BGP routing misbehavior against cyber-terrorism. In: IEEE Military Communications Conference (2005)
34. Smith, R.: The dynamics of internet traffic: self-similarity, self-organization, and complex phenomena. ArXiV:0806.3374 (2008)
35. Wang, L., Zhao, X., Pei, D., Bush, R., Massey, D., Mankin, A., Wu, S., Zhang, L.: Observation and analysis of BGP behavior under stress. In: Proceedings of the 2nd ACM SIGCOMM workshop of internet measurement (2002)
36. Wang, Y.: Protecting mission critical networks. Seminar on Network Security Publications in Telecommunications Software and Multimedia (2001)
37. Yuan, J., Mills, K.: Macroscopic dynamics in large-scale data networks. Complex Dynamics in Communication Networks (2005)
38. Zou, C.C., Gong, W., Towsley, D.: Code red worm propagation modeling and analysis. In: Proceedings of the 9th ACM conference on Computer and communications security (2002)
39. Zou, C., Gao, L., Gong, W., Towsley, D.: Monitoring and early warning for internet worms. In: Proceedings of the 10th ACM conference on Computer and communications security (2003)