Self-Organization of Dragon Kings

Yuansheng Lin,[1, 2, 3, *] Keith Burghardt,[3, 4] Martin Rohden,[3] Pierre-André Noël,[3] and Raissa M. D'Souza[3, 5, 6]

[1]*School of Reliability and Systems Engineering, Beihang University, Beijing, China, 100191*
[2]*Science and Technology on Reliability and Environmental Engineering Laboratory, Beijing, China, 100191*
[3]*Department of Computer Science, University of California, Davis, California, USA, 95616*
[4]*Department of Political Science, University of California, Davis, California, USA, 95616*
[5]*Department of Mechanical and Aerospace Engineering, University of California, Davis, California, USA, 95616*
[6]*Santa Fe Institute, Santa Fe, New Mexico, USA, 87501*

(Dated: November 13, 2017)

Outliers involving mechanisms absent in smaller events, known as Dragon Kings, are seen in many complex systems. An open problem, however, is whether models can self-organize to states that create Dragon Kings, or whether they only exist for certain externally fine-tuned parameter choices. In this Letter, we develop two related models where nodes in a network self-organize to be either "weak" or "strong" to failure when neighboring nodes fail. In one model, strong nodes are assumed to never fail, and the model displays self-organized criticality. However, in the other model, nodes are resistant, but not impervious, to the failure of multiple neighbors, and the model self-organizes to a state with Dragon Kings, which are highly sensitive to initial failures: a failure in a small portion of the network lead to global failures. We call this Dragon King mechanism a "runaway failure", in which the failure of subsystems piggyback off the failure of previous systems until almost all nodes fail. This model is in contrast with previous models where Dragon Kings occur in a subspace of parameters. We then demonstrate a simple control strategy that can decrease the frequency of Dragon Kings by orders of magnitude, which has been difficult to achieve for high-dimensional complex systems. Our models may give insight into self-organizing mechanisms underlying large-scale failures in natural and engineered systems.

PACS numbers: 89.75.Da 02.30.Yy 05.65.+b

Natural and engineered systems that usually behave in a manageable manner may nonetheless be prone to rare, catastrophic events [1–11]. Two categories have been proposed for these events: Black Swans, which are tail events in a power-law distribution, and Dragon Kings (DKs), which are outliers involving mechanisms absent in smaller events, and occur far more frequently than the power-law of a Black Swan would predict [1, 2]. The heavy-tailed distribution necessary for Black Swans to exist is often explained by self-organized criticality (SOC): a tug-of-war that poises the system close to a critical point without any need for external interventions [5–8]. Although prediction of Black Swans can sometimes beat random chance [12], the task appears to be inherently difficult [13]. Despite this drawback, there are simple methods to push SOC systems away from criticality, thus reducing the size of Black Swans [8, 14, 15].

It has been argued that DKs occur in complex systems that have low heterogeneity and strong coupling (as defined in [16]) and that, in contrast, Black Swans occur in systems with weaker coupling and higher heterogeneity. Whereas Black Swans have no associated length- and time-scales, DK events do: there are typical places and times when DKs will and will not occur. This has been successfully applied to, for example, prediction of material failure and crashes of stock markets [1], and has been seen in engineered systems, such as error cascades in a collection of robots [17]. Unlike Black Swans, however, it has been an open problem to control DKs in many situations and to elucidate the mechanisms underlying these self-amplifying cascades [2]. Recent advances to control DKs have been based on low-dimensional models, such as coupled oscillators [18], but control of DKs in models of high-dimensional complex systems has been lacking.

In this Letter, we strive to capture the essence of Black Swan and DK formation on complex systems through two related self-organizing network models. Both consist of nodes that are either "weak" or "strong" in the face of neighboring failures. These models are identical in most aspects: a weak node fails as soon as *one* of its neighbors fails, a failed weak node has small probability, ϵ, to be reinforced and upgraded to a strong node, and strong nodes independently degrade (i.e., become weak) at a slow rate. The two models differ, however, in the strong-node failure-spreading mechanisms: either strong nodes *cannot* fail, which we call *inoculation* (IN) [19], or they fail as soon as *two* of their neighbors fail, which we call *complex contagion* failure (CC) [20]. The former model is alike to site percolation, and is the null model we compare the CC model to.

We are interested in the long-term behavior: each cascade causes small changes in the number of weak and strong nodes, which induces the slow self-organization of the network. As the reinforcement probability, ϵ, approaches zero, both models self-organize to specific (but distinct) states. While the IN model is similar in spirit to some previous self-organized critical models of engineered systems [7, 8], the CC model spontaneously generates DKs (failures of nearly the entire system) over all values of $\epsilon < 1$, a prediction that we have confirmed for ϵ over several orders of magnitude. These DKs are fundamentally different than previous models of self-organized systems, where the models were externally tuned to create DKs [1], and those DKs did not affect the entire system. The main reason for this significant difference is that DK failures in the CC model tend to occur when the first cluster of weak nodes to fail is sufficiently larger than a certain value. This is related to the probability that a strong node that bridges weak-node clusters will have two neighbors in the initial failing cluster, a generalization of the birthday problem [21]. Once this probability is significant, then failures are likely to spread from the first weak-node cluster to subsequent weak-node clusters. More strong nodes are then likely to fail by piggybacking off of the previous failures. We can make an qualitative analogy to the gas-water phase transition in condensed matter, where water droplets (the failure size) have a surface tension (analogous to the strong nodes) which makes larger droplets energetically dis-favorable, while the inside of the water droplet (analogous to the the cluster of failed nodes) makes larger droplets (larger failures) energetically favorable. In both our model and in droplet nucleation, there is a critical size, above which the droplet or failed cluster grows almost without bound [22].

We take advantage of this finding to predict whether a small initial failure will cascade into a DK event. Moreover, a simple targeted-reinforcement control strategy, in which we turn a few fairly well-chosen weak-nodes into strong nodes, can decrease the likelihood of DKs and other large failures by orders of magnitude.

Self-organizing models. The dynamics of our models depend on two competing mechanisms: degradation and reinforcement. Degradation, which represents the aging of infrastructure or an increase of load placed on them, is modeled by slowly converting strong nodes into weak ones. Conversely, reinforcement converts weak nodes that fail during a cascading event into strong nodes at rate ϵ, representing the hardening of nodes in an attempt to prevent future failures. This is a reasonable assumption mimicking modern-day power grid guidelines [23], where resources are allocated to places were failures happen more often. The trade-off between degradation and reinforcement drives the system to an SOC state.

For simplicity, we consider dynamics on a 3-regular random networks with N nodes, where N is an even positive integer. Repeated edges and self-loops are allowed, but are rare when N is large. The system size and the probability ϵ are the model's only parameters. We are particularly interested in large N and small ϵ, but due to motivation from real-world systems, we are also interested in finite-size effects as well as the consequences of a non-zero ϵ.

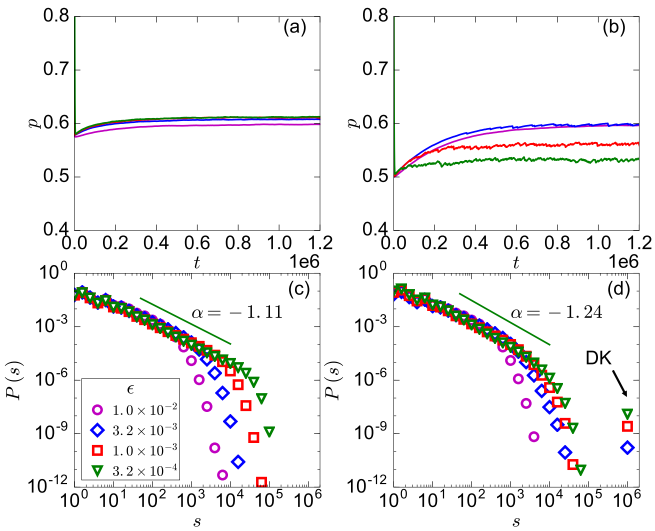

FIG. 1. (Color online) Self-organizing behavior and failure size. Top row: Fraction of weak nodes $p(t)$ vs. t for the (a) IN and (b) CC models over individual network realizations for $N = 10^6$. Larger values of ϵ cause slower relaxation. Bottom row: Failure size distribution for the (c) IN and (d) CC models, with the DK outliers labeled. Symbols denote results of simulations on random 3-regular graphs averaged over ten network realizations and $15 \times N$ time steps.

Both the CC and IN models follow the same general algorithm. At time $t = 0$, we initialize all N nodes as weak. Weak nodes fail if at least one of their neighbors fail which can cause subsequent failures. The distinction is that under the CC model, strong nodes fail if at least two of their neighbors fail, whereas in the IN model, strong nodes cannot fail. In detail, each discrete time step $0 \leq t \leq t_{\text{stop}}$ proceeds according to the following algorithm.

Degradation: Select a node uniformly at random. If that node is strong, make it weak and proceed to the beginning of the Degradation step with $t \leftarrow t+1$. If the selected node is already weak, then it fails, and continue with the remaining three steps.

Cascade: Apply the IN or CC failure-spreading mechanism until no more failures occur. Failed nodes remain failed for the duration of the cascade.

Repair: All failed nodes are un-failed (strong failed nodes become strong un-failed nodes, and weak failed nodes become weak un-failed nodes).

Reinforcement: Each weak node that failed at this time step has probability ϵ to become strong. Proceed back to the Degradation step with $t \leftarrow t+1$.

Many other choices for initial conditions are possible, but our investigations show that the steady state behavior is independent of these choices (see SI). Because we currently initialize all nodes as weak, the sizes of the first few cascades are on the order of the system size, and numerous node upgrades take place before the system equilibrates. An important indicator that we have reached the relaxation time is the proportion of nodes that are weak at time t, $p(t)$, which is shown in the top row of Fig. 1. We wait until well after $p(t)$ stabilizes ($5 \times N$ timesteps) and then calculate failure sizes for a subsequent $15N$ timesteps. Although we cannot prove that the model has reached equilibrium, waiting longer, and varying the initial conditions (see SI) produces quantitatively similar results. For the IN model, we find that $p(t)$ is almost independent of ϵ as ϵ approaches zero, but in the CC model, the steady-state value of $p(t)$ depends on ϵ.

Failure size distribution. The results for the failure size distribution, $P(s)$, are illustrated in the bottom row of Fig. 1, which demonstrates each model's propensity to create large failure events. The probability of large failures generally increases with decreasing ϵ for both the IN and CC models because, if less nodes are reinforced, cascades can more easily spread and affect larger portions of a network. For small enough ϵ, we find that the cascade size distributions for the IN and CC models exhibit a power-law with exponential decay, however the CC model also has a DK tail, where *over 99.9% of nodes fail* in each DK event (cf. Fig. 1(d)). Furthermore, they appear to have two different power-law exponents: $\alpha = -1.11$ for the IN model and $\alpha = -1.24$ for the CC model when $\epsilon = 3.2 \times 10^{-4}$ and $N = 10^6$ (in comparison, traditional SOC models yield $\alpha = -1.5$ [24, 25]).

Dragon King Mechanism. Why do DKs occur in the CC model? To establish a theoretical understanding of DKs, we first note that a failure in any part of a weak-node cluster makes the entire cluster fail. A necessary, but not sufficient, condition for a DK to occur is that strong nodes bridging the first failed weak-node cluster must also fail (cf. Fig. 2(a)), which we call it a one-step failure cascade. In the simplest case, only one strong node bridges two weak-node clusters. We first analyze the probability that the failure of a weak-node cluster, with size $C_{\mathrm{w},1}$, will lead to the failure of at least one bridging strong node, denoted by S_1, and find that S_1 nodes can accurately model the probability of multiple weak-node clusters failing (see SI).

The one-step failure cascade is, however, a poor approximation of a DK event (cf. Fig. 2(b)), where cascades leads to yet more cascades (i.e., failed weak-node clusters lead to subsequent cluster failures until almost all nodes fail). To better understand DKs, we need to know whether the first failed weak-node clusters will lead to further failures, e.g., a two-step cascade, as seen in S_2 in Fig. 2(a). To obtain this probability, we first prove that the number of weak-node clusters that fail just after the first weak-node cluster fails is Poisson distributed. Furthermore, if we assume that the clusters are independent and identically distributed random variables from a scale-free distribution, then we can find the distribution of failed weak nodes after the first step of a cascade, which we use to calculate the probability of two-step failure cascades (a necessary condition for DKs), given the initial cluster size $C_{\mathrm{w},1}$. We numerically find that $P(\text{two-step cascade}|C_{\mathrm{w},1}) = 1/2$ when

$$C_{\mathrm{crit},2} \sim N^{0.55\pm0.01}. \quad (1)$$

Although a necessary condition, a two-step failure cascade does not always create a DK, therefore $P(\text{two-step cascade}|C_{\mathrm{w},1}) > P(\mathrm{DK}|C_{\mathrm{w},1})$, which implies that $C_{\mathrm{crit},2} < C_{\mathrm{crit,DK}}$, where $C_{\mathrm{crit,DK}}$ is the critical size of $C_{\mathrm{w},1}$ such that $P(\mathrm{DK}|C_{\mathrm{w},1}) = 1/2$. We find that these bounds agree with what we see numerically (cf. Fig. 2(b)), and $C_{\mathrm{crit},2}$ is in much closer agreement than $C_{\mathrm{crit},1}$ is to $C_{\mathrm{crit,DK}}$. We next consider how these critical values scale with system size. Equation (1) implies that $O(N^{0.55}) \leq C_{\mathrm{crit,DK}} \leq N$. These bounds are in agreement with the numerical scaling, in which $C_{\mathrm{crit,DK}} \sim N^{\gamma}$, where $\gamma = 0.59 \pm 0.03$ (see SI). Importantly, $C_{\mathrm{crit,DK}}$ scales sub-linearly, therefore only a small proportion of the network needs to initially fail before a DK is likely to occur.

Finally, we can discuss how $P(\mathrm{DK})$ varies with N and ϵ. First, we make simplifying assumptions that any $C_{\mathrm{w},1} > C_{\mathrm{crit,DK}}$ creates a DK, and approximate $P(C_{\mathrm{w},1})$

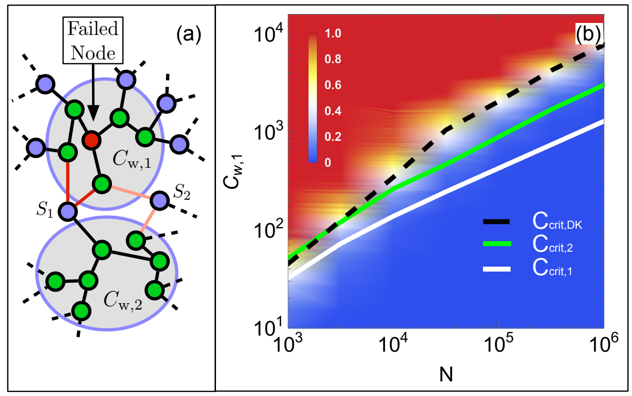

FIG. 2. (Color online) DKs form by cascading failures of weak-node clusters. (a) Weak-node clusters (circled) are surrounded by strong nodes. If one such strong node (S_1) has two links connecting to the same weak-node cluster, the failure of this cluster (with size $C_{\mathrm{w},1}$) will make the strong node fail, and the cascade may spread to other weak-node clusters (e.g., one of size $C_{\mathrm{w},2}$), and thus other strong nodes (e.g., S_2), eventually creating a DK event. (b) A heat map of the probability a DK occurs in the CC model conditioned on $C_{\mathrm{w},1}$, the size of the weak-node cluster that first fails, versus N and $C_{\mathrm{w},1}$ for $\epsilon = 10^{-3}$. Black dashed line is the simulation result for $C_{\mathrm{crit,DK}}$, while solid lines denote our analytic calculations of two-step failure cascades $C_{\mathrm{crit},2}$ (green line) and one-step failure cascades $C_{\mathrm{crit},1}$ (white line).

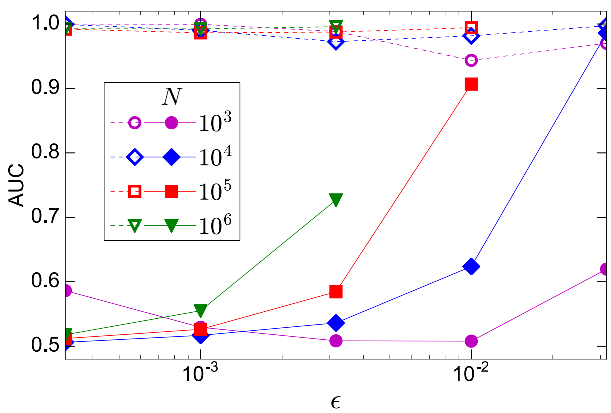

FIG. 3. (Color online) Predicting DKs. The area under the receiver operating characteristic (AUC) for logistic models of $P(\mathrm{DK}|p)$ (closed symbols and solid lines) and $P(\mathrm{DK}|C_{\mathrm{w},1})$ (open symbols and dashed lines) for varying N [26].

as a power-law with an exponential cut-off, λ, that is proportional to ϵ (see SI). We find that increasing ϵ by a small amount creates an unexpectedly large percentage reduction of $P(\mathrm{DK})$ as well as large percentage reduction in failures that are not DKs (cf. Fig. 1). Surprisingly, DKs exist for any value of ϵ. If ϵ is proportional to the cost of upgrades, then our results suggest that upgrading failed components in a system slightly more frequently can dramatically reduce the probability of serious failures. When $\epsilon \to 0$ and N is large, the theory suggests that $P(\mathrm{DK}) \sim N^{-0.15\pm0.02}$ (see SI), therefore DKs slowly disappear in the thermodynamic limit, but the scaling exponent is so small, that DK events are visible for almost any value of N and ϵ.

Predicting Dragon Kings. DKs are, in contrast to Black Swans [12, 13], fairly predictable [1], although it may not be obvious what independent variables best indicate these events. For example, we find little correlation in the time between DKs (the autocorrelation is < 0.01 for $N = 10^6$, see SI), therefore, knowing the time-series of DKs will not tell is when another will necessarily occur. To answer this question, we analyze two different predictors. The first predictor is the fraction of weak nodes present in the network. The rational is that more weak nodes create larger initial failures, and therefore create more DKs. The second predictor is the size of the first weak-node cluster, $C_{\mathrm{w},1}$. We earlier established that the probability of DKs correlates with $C_{\mathrm{w},1}$, although we have yet to see whether this is adequate for predicting DKs. Both of these predictors are complimentary, because the former would tell us *when* a DK might occur, while the latter would tell us *where* a DK might originate, i.e., whether a cascade in progress will lead to a DK.

We model $P(\mathrm{DK}|p)$ versus p, and $P(\mathrm{DK}|C_{\mathrm{w},1})$ versus $C_{\mathrm{w},1}$, respectively, using logistic regression. Unless ϵ is relatively large, p is a poor predictor as based on the area under the receiver operating characteristic (AUC, cf. Fig. 3) [26]. Thus, predicting when a DK would occur is inherently challenging. In contrast, by knowing $C_{\mathrm{w},1}$ alone, we can predict DKs with astounding accuracy, almost independent of N and ϵ. The high accuracy is due to the characteristic size of the initial failure that triggers a DK, $C_{\mathrm{crit,DK}}$ (see SI for figure). This is reminiscent of previous results on controlling DKs in a system of oscillators where a trajectory straying past a particular threshold is very likely to create a DK [18, 27]. For each node, finding $C_{\mathrm{w},1}$ requires searching locally in the network until we find neighboring strong nodes, and because $C_{\mathrm{w},1} \ll N$, the effort this would require is small, therefore, given an initial failure, we can accurately predict whether a DK would occur with relatively little effort. Similarly, to "tame" DKs, we can use a simple control mechanism that requires knowing the size of just a few weak-node clusters, as seen in the next section.

Controlling Dragon Kings. Because large weak-node cluster failures precede DKs, we can reasonably ask whether breaking up these clusters before they fail can reduce the prevalence of DKs. Assuming that the rate of node upgrades is proportional to the amount of "money" or effort allocated for repairing nodes, we create control strategies where this rate is kept the same on average as the non-controlled case, meaning $p(t)$ remains approximately constant. Instead of randomly reinforcing failed weak nodes, we upgrade weak nodes in large clusters by picking r weak nodes and finding the size of the weak-node clusters to which they belong. The largest of these weak-node clusters is selected and with probability $1 - p(t)$, a random node in that cluster is reinforced. We find that, when $r = 1$, more DKs occur than without control therefore some attempts to reduce the size of failures could actually could make the failures worse. However, larger r represents a better sampling of the cluster sizes, and a greater chance for large clusters to be broken apart, which reduces the probability of DKs by orders of magnitude (cf. Fig. 4(a)), as well as large failures that are not DKs (cf. Fig. 4(b)). Furthermore, the number of nodes we have to search through is only $r \times \langle C_{\mathrm{w},1} \rangle \ll N$ on average, which makes this technique applicable in systems where global knowledge of the network is lacking.

Discussion. We have shown that DKs can self-organize in the CC model via runaway failure cascades. Moreover, this mechanism allows for DKs to be easily predicted and controlled. We believe that this model can describe a number of mechanisms, discussed below.

The CC model allows for individuals with simple contagion dynamics (weak nodes) [28], and complex contagion dynamics (strong nodes) [29], to co-exist on a network and assumes that agents become "complex" at a rate ϵ after they have adopted an idea (failed), which can be interpreted as agents exhibiting greater stubbornness to new ideas. The CC model suggests that agents can

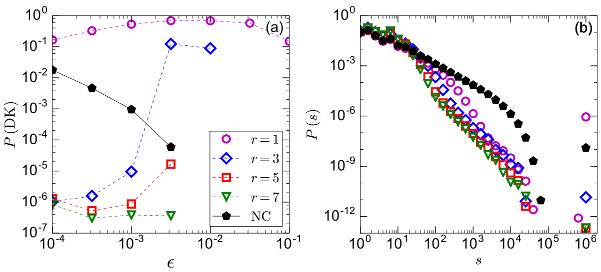

FIG. 4. (Color online) Controlling DKs. (a) The probability a DK occurs over time versus ϵ in both the non-controlled scenario (NC, pentagons), and in the controlled scenario with r weak-nodes chosen: $r = 1$ (circles), $r = 3$ (diamonds), $r = 5$ (squares) and $r = 7$ (triangles). Simulations are realized for $N = 10^6$, and standard errors are smaller than marker sizes. See main text for details of the control method. (b) Failure size distributions for different control strategies and non-controlled with $\epsilon = 3.2 \times 10^{-4}$.

self-organize to a state in which global adoption (DKs) occurs surprisingly often. This could explain, for example, the mechanism of large financial drawdowns in stock markets, which are found to be DKs [1], where social interactions, seen in stock market participation [30] and foreign exchange trading [31], can convince brokers to buy or sell as a group. Some brokers will buy (sell) stock when any neighbor does, while other agents buy (sell) stock only after multiple neighbors do. Our model may also represent mechanisms for cascading failures seen, for example, in electrical power grids [32, 33], where reinforcement of failed units (represented as nodes in our model) is a common practice [23]. Nodes, representing a part of a complex system, can degrade and be reinforced at slow rates to represent upgrade costs that are high and often limited to the point of making the system barely stable [7, 8]. Surprisingly, however, we find that reinforcing a system slightly more often, or selectively reinforcing nodes (the control strategy with $r > 3$), creates a significant percentage drop in the frequency of DKs. In contrast, naively reinforcing nodes at random (the control strategy with $r = 1$) dramatically increase the frequency of DKs.

The IN and CC models provide novel mechanisms for cascading failures. The IN model can help explain why a failure size distribution in a complex system follows a power-law, which is seen, for example, in electrical blackouts [7, 34], while the CC model can help explain why DKs exist in the failure size distribution of real systems. Although a fundamental assumption of the IN model is that strong nodes never fail, it is reasonable to expect that reinforced nodes in real systems can fail, therefore the CC model may describe the mechanism behind many cascading failures.

Future work is necessary to verify that certain real systems have analogous failure mechanisms, and to better understand the mechanism of DKs. The research presented here, however, provides a concrete methodology to begin studying how DKs are driven by the interplay of heterogeneity (for example proportion of strong nodes) and coupling (e.g., node degree) in a principled manner, which is still in its infancy [1]. We have also not explored the effect that the mean degree, the degree distribution, or the failure threshold has on CC dynamics. Generalizing the CC model to these networks also creates additional degrees of freedom, for example the failure dynamics could depend on the minimum number of neighboring agents [20], or minimum *fraction* of neighboring agents [35], that need to fail for the failure to spread to a strong node. This distinction becomes important for heavy-tailed degree distributions.

We gratefully acknowledge support from the US Army Research Office MURI award W911NF-13-1-0340 and Cooperative Agreement W911NF-09-2-0053; The Minerva Initiative award W911NF-15-1-0502; and DARPA award W911NF-17-1-0077; and financial support from the program of China Scholarships Council (No. 201506020065, Y.L.).

* bhlys1106@gmail.com

[1] D. Sornette, International Journal of Terraspace Science and Engineering **2**, 1 (2009).
[2] D. Sornette and G. Ouillon, Eur. Phys. J. Special Topics **205**, 53 (2012).
[3] J. M. Carlson and J. Doyle, Phys. Rev. E **60**, 1412 (1999).
[4] P. Bak, *How Nature Works: the Science of Self-organized Criticality* (Copernicus, New York, 1996).
[5] P. Bak, C. Tang, and K. Wiesenfeld, Phys. Rev. Lett. **59**, 381 (1987).
[6] P. Bak, C. Tang, and K. Wiesenfeld, Phys. Rev. A **38**, 364 (1988).
[7] B. A. Carreras, V. E. Lynch, I. Dobson, and D. E. Newman, Chaos: An interdisciplinary journal of nonlinear science **12**, 985 (2002).
[8] H. Hoffmann and D. W. Payton, Chaos Soliton. Fract. **67**, 87 (2014).
[9] S. Wheatley, B. Sovacool, and D. Sornette, Risk analysis **37**, 99 (2017).
[10] J. Lorenz, S. Battiston, and F. Schweitzer, The European Physical Journal B-Condensed Matter and Complex Systems **71**, 441 (2009).
[11] C. J. Tessone, A. Garas, B. Guerra, and F. Schweitzer, Journal of Statistical Physics **151**, 765 (2013).
[12] O. Ramos, E. Altshuler, and K. J. Måløy, Physical review letters **102**, 078701 (2009).
[13] R. J. Geller, D. D. Jackson, Y. Y. Kagan, and F. Mulargia, Science **275**, 1616 (1997).
[14] D. O. Cajueiro and R. F. S. Andrade, Physical Review E **81**, 015102 (2010).
[15] P.-A. Noël, C. D. Brummitt, and R. M. D'Souza, Phys. Rev. Lett. **111**, 078701 (2013).
[16] I. Osorio, M. G. Frei, D. Sornette, J. Milton, and Y.-C. Lai, Physical Review E **82**, 021919 (2010).
[17] M. Gauci, M. E. Ortiz, M. Rubenstein, and R. Nagpal,

in *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)* (International Foundation for Autonomous Agents and Multiagent Systems, 2017) pp. 1404–1412.
[18] H. L. d. S. Cavalcante, M. Oriá, D. Sornette, E. Ott, and D. J. Gauthier, Physical review letters **111**, 198701 (2013).
[19] W. O. Kermack and A. G. McKendrick, Proc. R. Soc. A: Math. Phys. Eng. Sci. **115**, 700 (1927).
[20] D. Centola and M. Macy, American journal of Sociology **113**, 702 (2007).
[21] A. O. Allen, *Probability, statistics, and queueing theory* (Academic Press, 2014).
[22] X. C. Zeng and D. W. Oxtoby, J. Chem. Phys. **94**, 4472 (1991).
[23] T. A. Short, *Electric Power Distribution Handbook* (CRC Press, Boca Raton, Fl, 2004).
[24] P. Alstrøm, Phys. Rev. A **38**, 4905 (1988).
[25] K. Christensen, H. Flyvbjerg, and Z. Olami, Physical review letters **71**, 2737 (1993).
[26] C. D. Brown and H. T. Davis, Chemometrics and Intelligent Laboratory Systems **80**, 24 (2006).
[27] A. E. Motter, Physics **6**, 120 (2013).
[28] N. Hodas and K. Lerman, Sci. Rep. **4** (2014), 10.1038/srep04343.
[29] D. Centola, Science **329**, 1194 (2010).
[30] H. Hong, J. D. Kubik, and J. C. Stein, J. Finance **59**, 137 (2004).
[31] W. Pan, Y. Altshuler, and A. Pentland, in *International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing* (IEEE, 2012) pp. 203–209.
[32] M. Rohden, D. Jung, S. Tamrakar, and S. Kettemann, Phys. Rev. E **94**, 032209 (2016).
[33] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, Chaos: An Interdisciplinary Journal of Nonlinear Science **17**, 026103 (2007).
[34] X. Weng, Y. Hong, A. Xue, and S. Mei, Journal of Control Theory and Applications **4**, 235 (2006).
[35] D. J. Watts, Proceedings of the National Academy of Sciences **99**, 5766 (2002).