Higher-order interaction learning of line failure cascading in power networks

A. Ghasemi[1] and H. Kantz[2]

[1])*Computer Engineering Department, K. N. Toosi University of Technology, 1631714191 Tehran, Iran*

[2])*Max Planck Institute for Physics of Complex Systems, Nöthnitzer Str. 38, 01187 Dresden, Germany*

(*Electronic mail: arghasemi@kntu.ac.ir)

Line failure cascading in power networks is a complex process that involves direct and indirect interactions between lines' states. We consider the inverse problem of learning statistical models to find the sparse interaction graph from the pairwise statistics collected from line failures data in the steady states and over time. We show that the weighted l_1-regularized pairwise maximum entropy models successfully capture pairwise and indirect higher-order interactions undistinguished by observing the pairwise statistics. The learned models reveal asymmetric, strongly positive, and negative interactions between the network's different lines' states. We evaluate the predictive performance of models over independent trajectories of failure unfolding in the network. The static model captures the failures' interactions by maximizing the log-likelihood of observing each link state conditioned to other links' states near the steady states. We use the learned interactions to reconstruct the network's steady states using the Glauber dynamics, predicting the cascade size distribution, inferring the co-susceptible line groups, and comparing the results against the data. The dynamic interaction model is learned by maximizing the log-likelihood of the network's state in state trajectories and can successfully predict the network state for failure propagation trajectories after an initial failure.

Line failure cascading in power grid networks involves higher-order interactions causing complex dynamics in failure cascading. The pairwise line failure interaction analysis indicates to what extent the failure of one line may lead to overload or failure of another line and has been studied in deterministic and stochastic frameworks. Here, we show that the cascading process involves higher-order interactions of groups of more than two lines whose simultaneous states affect process dynamics. Nevertheless, direct data collection and analysis of all possible combinations of lines' states in different group sizes is impossible due to the explosive number of groups. Therefore, we use machine learning techniques and prior knowledge to find a statistical model which captures direct and possible indirect higher-order interactions in the complex dynamics of failure cascading. We use the learned models to infer the cascade behavior and compare the results against the data.

––––––––––––––––––

I. INTRODUCTION

The failure cascading process is a high-risk event in networked systems in which the overall cost, e.g., the number of shutdown users in the power grid, increases in the same order as the probability of the event decreases. In networked systems, the direct and indirect interactions between the system components induce correlations and may amplify or attenuate the initial disturbance. The amplification[1] or attenuation[2] effects of network structure after especially correlated fluctuations reflect the underlying interplay between the structure and dynamics of the complex networked systems.

In power networks, lines' failure cascading are correlated in a non-trivial pattern, rarely leading to large blackouts accord-ing to the historical data[3]. The origins of cascading process in power networks are related to the self-organized critical-ity phenomenon in complex systems in[4,5] and, more recently, are linked to the power-law nature of city inhabitants[6]. Some other studies, instead of finding what gives rise to the phe-nomenon, focus on finding how the cascade process relates to the network's structure and how it unfolds in the network in a deterministic[7] or stochastic manner[8]. These studies link the failure unfolding process to the pairwise line interaction. The pairwise line interaction refers to the mutual impact that a pair of lines has on each other after a failure of one of them.

Provided that the network remains connected, the authors in[7] use the deterministic pairwise line outage redistribution factors (LODFs) and matrix-tree theorem to analyze how fail-ure propagates through spanning forests in the network graph. However, the network partitions into some islands in many failure cascading scenarios. On the other hand, a class of data-driven approaches rely on analyzing positive pairwise line in-teractions statistics after different initial failure scenarios[8,9]. As we shall discuss, positive pairwise statistics like pairwise correlations do not capture some crucial interactions, and neg-ative interactions play a crucial role in cascading process.

Another class of data-driven approaches considers multiple line failure interactions in the consecutive generation of fail-ure cascading in the networks by analyzing how a group of line failures may affect the failure of a specific component. In[10] by assuming a memory between consecutive generations, the evolution of interactions during the cascade unfolding is analyzed, and an interaction matrix for two consecutive gener-ations is derived. In[11], the set of lines which are failed in a spe-cific generation of cascading data are considered as the states of a Markov chain, and the transitions between these states are estimated. The Markov states can transit to the consid-ered stopping state that terminates the cascade. The simulated cascades using the developed models in these works capture

the distribution of the number of failures in the original data. See[12] for a review on interaction analysis of failure cascading in power networks. In this work, we use different methods to find the line failure interactions near the cascades' final states and, over time, use the data. We use machine learning tools and pair-wise statistics to capture the interaction between a possible group of lines without relying on specific assumptions about the interaction between cascading generations or enumerating the Markov states.

Although finding the pairwise statistics is straightforward and computationally tractable even for large networks, they are not sufficient per se if higher-order interactions exist. Despite the pairwise interaction, in higher-order interactions, the simultaneous states of more than two lines are involved in determining the system dynamics. Higher-order interactions may substantially affect the dynamics of complex networked systems[13]. The failure cascading process in power grid networks involves higher-order interactions, as we discussed in more detail in subsection Ref sec: higher-order. However, collecting data for possible higher-order interactions is not straightforward, if even possible, due to the explosive number of possible combinations. Therefore, there is an interest in finding the possible higher-order interaction using ordinary pairwise statistics. Pairwise models assume that the response of each element in the networked system results from its pairwise interactions with some not-necessarily local elements. The efficiency of the pairwise statistical model to capture higher-order correlations was first observed in the study of strongly correlated network states of neural activity dynamics in[14]. Also, Ref.[15] shows that the Pseudo-likelihood and approximate maximum entropy statistical model can successfully recover the interaction topology even from a limited amount of data.

In this paper, we consider the inverse problem of learning the interaction graph from the pairwise statistics collected from data of line failures in the steady states and over time. After presenting the system model in Section II, we discuss that the failure cascading process in power grid networks involves higher-order interactions undistinguished by observing the pairwise correlation data in Section III. Next, we aim to learn statistical models that capture the latent higher-order line failure interactions. The models use ordinary pairwise statistics data to successfully predict complex system responses like the cascade size statistics and consecutive network states. We find static and dynamic interaction graphs in Section IV and Section V. The static interaction graph helps us estimate the cascade size distribution and identify lines that fail together. On the other hand, the time series analysis helps find how the failure unfolds in the network.

II. MODELS AND DATA SET PREPARATION

A. System model

Consider a power grid network with $\mathcal{N} = \{1, \ldots, n\}$ buses or nodes and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}, |\mathcal{E}| = L$, transmission lines or edges with the corresponding graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. In the normal operation, the network facilitates the electricity flow distribution from generator buses to load buses meeting the underlying system's physics (Ohm's rule, flow conservation rule, and power balance) and its constraints, i.e., the maximum generation power of generators and the maximum capacity of lines.

Ignoring the lines' resistances, the susceptance of line $e = (i,j) \in \mathcal{E}$ between bus i and j is given by $b_{ij} = \frac{1}{x_{ij}}$ where x_{ij} is the line's reactance. Let $\mathbf{B}_{L \times L} = \mathrm{diag}(b_e : e \in \mathcal{E})$ and $\mathbf{C}_{n \times L}$ denote, respectively, the susceptance and the node-link incidence matrix of $\mathcal{G}$ assuming an arbitrary orientation for each link. In this paper, all matrices and vectors are, respectively, denoted by bold uppercase and bold lowercase letters. The power injection or demand at bus i is p_i and $\mathbf{p}_{n \times 1} = (p_1, \ldots, p_n)$ is the corresponding vector. f_e is the flow on link e and $\mathbf{f}_{L \times 1} = (f_1, \ldots, f_L)$ is the flow vector of the network. Assume that the voltage magnitude of all buses is normalized to 1 and the unknown voltage phase of bus i is denoted by θ_i. In the linear model, applying Ohm's law for link $e = (i,j)$ we have $f_e = (\theta_i - \theta_j)b_{ij}$, which in the matrix form reads as $\mathbf{f}(t) = \mathbf{B}(t)\mathbf{C}(t)^T \boldsymbol{\theta}(t)$. The flow conservation law at each bus meets, $\mathbf{C}(t)\mathbf{f}(t) = \mathbf{p}(t)$. Ohm's law and flow conservation, along with the power balance constraint $\mathbf{1}^T \mathbf{p}(t) = 0$, ends up to finding $n-1$ unknown voltage phases assuming the voltage phase of the slack bus generator as zero. The power of the slack bus adjusts to meet the small fluctuations in the power supply-demand balance in the network. Specifically, let $\mathbf{L}(t) = \mathbf{C}(t)\mathbf{B}(t)\mathbf{C}^T(t)$ denote the Laplacian matrix of the $\mathcal{G}$, i.e., $L_{ij} = -b_{ij}$ if there is a link between i and j and $L_{ii} = \sum_j b_{ij}$. The voltage phases are then given by $\boldsymbol{\theta}(t) = \mathbf{L}^{\dagger}(t)\mathbf{p}(t)$ where $\mathbf{L}^{\dagger}$ is the Moore-Penrose inverse of $\mathbf{L}$. Finally, using Ohm's law the flow of each line reads as $\mathbf{f}(t) = \mathbf{B}(t)\mathbf{C}(t)^T \mathbf{L}^{\dagger}(t)\mathbf{p}(t)$.

Each generator has a capacity above which it will shut down. Also, there is a capacity for line e, c_e, and the line fails if its flow exceeds its capacity. Therefore, the steady-state lines' flows are the solutions of the above linear model subject to many physical constraints. The network is subject to line failure perturbations in time, e.g., due to lightning or malfunctioning of relays. After the initial failure, the flows are redistributed. This may lead to subsequent line failures, power imbalance, and even network partitioning before the network settles into a new steady state. This linear flow distribution and redistribution model in the power grid captures essential features of the cascade process like a non-additive response, non-local propagation, and disproportional impact[16] and is used in other works[8,17].

B. Data set preparation

We develop a simulator to collect a data set of failure cascading trajectories for given network topology, power generation and demands at buses, the maximum power of generators, and the capacity of lines.

The initial flow of each line is computed using the flow distribution model, assuming all lines are working properly. At each run, the process starts with randomly removing a small

random subset of lines in which each line is removed independently with probability p_f. Next, the new line flows are recomputed, and if a line's flow exceeds its corresponding capacity, that line fails as well, which may trigger other consecutive failures. We record the failed lines at each time step until the network settles at a steady state. The network may disconnect due to failures and decompose into components. Therefore, the power balancing of the network or its components may be destroyed. We adopt the power re-balancing strategy explained in[18]. In this strategy, the small power imbalance is compensated by ramping up or ramping down the power generation at generators. Beyond that, we use generator tripping and load shedding with the priority of small generators or loads. We simulate and collect M trajectories of failure cascading on the IEEE-118-Bus network. The required data, including the network connectivity, the lines' capacities, and the maximum generators' powers, are available in[19]. The basic statistics of IEEE-118-Bus network are $N = 118$, $L = 179$; mean degree $\langle k \rangle = 3.034$; clustering coefficient $C = 0.136$.

We perform our experiments on two data sets. We set $p_f = \frac{2.5}{L}$ in our data collection phase. The first data set D_1 consists of $M \approx 52000$ unique trajectories with random initial failure scenarios. Due to the available redundancies, many initial failures do not propagate. In this data set, 46% of the initial failures lead to at least one consecutive failure, while the remaining 54% do not propagate. This data set is used to infer the interactions in the normal operation of the network. Data set D_2 consists of about $M \approx 38000$ trajectories in which all of the initial failures propagate at least one step. The learned interaction matrix from this data highlights the indirect interactions in the cascading scenarios.

In the following the state of line i, is denoted by $s_i = \pm 1$, where $s_i = +1$ indicates that the line fails. The state of network is completely determined by $\mathbf{s}(t) = (s_1, \ldots, s_L)$. We measure the cascade size, Z, in terms of the number of failed lines, $Z = \sum_{i=1}^{L} \frac{(1+s_i)}{2}$. Note that, although the details of simulations like the power balancing strategy affect the collected data sets, the main interesting feature of observing heavy tail distribution in the cascade size remains unchanged. We are interested in exploiting these data to learn statistical models which encode the lines' interactions and use them to infer lines that fail together, the influential lines, and how the cascade unfolds in time.

III. PAIRWISE AND HIGHER-ORDER INTERACTIONS

This section first explains the pairwise line failure interaction in the power network. We use this prior knowledge in our learning schemes. Next, we discuss possible higher-order interactions that might be undistinguished by directly observing the pairwise correlations.

A. Pairwise interactions

For a given pair of lines, the (asymmetric) pairwise interaction shows to what extent one line's failure may lead to consecutive overload or failure of the other line. Let (a,b) denote the line between nodes a and b and consider the pair $e = (a,b)$ and $\hat{e} = (c,d)$. Assume e fails. The line outage redistribution factor (LODF), $K_{e\hat{e}}$, is the ratio of flow changes on line $\hat{e}$ to the initial flow on line e before it was failed provided that the network remains connected. $K_{e\hat{e}}$ is independent of the power injection or demand vector $\mathbf{p}$ and only depends on the underlying weighted graph and can be efficiently computed deterministically[7].

Specifically, $K_{e\hat{e}}$ depends on the weight of certain spanning forests in the graph $\mathscr{G}$. In particular, if e and $\hat{e}$ are connected to a common bus we have $K_{e\hat{e}} > 0$. That is, the proximity in the physical network usually implies interactions as we expected. Alternatively, one could find the pairwise line failure correlations using a reasonable amount of recorded data or simulation.

Statistical analysis of the spatial spreading of line failure cascading based on observed utility data shows that as the network distance between two lines increases, the probability of successive failure is decreased[20]. The same conclusion is also drawn for the line failure cascading based on the flow redistribution model in[21]. Note that we observe physically far distance but strong interacting line pairs. We use this prior knowledge to adjust the regularization (penalization) factor in learning the interaction structure between lines assuming that the farther the network distance between the lines, the less strong the interaction value is. We adopt the edge distance, $d_{e,\hat{e}}$, which was introduced in[21] to investigate the non-local effect of failure cascading. Let $d_{x,y}$ denote the shortest path length between nodes x and y in $\mathscr{G}$. We have $d_{e,\hat{e}} = \min_{x \in \{a,b\}, y \in \{c,d\}} d_{x,y} + 1$. Note that if e and $\hat{e}$ are connected to a common bus $d_{e,\hat{e}} = 1$.

B. Higher-order interactions

Pairwise statistics of lines' failures are not sufficient per se for cascade process analysis if the process involves higher-order interactions. Higher-order interaction refers to a group of more than two lines whose simultaneous states affect system dynamics.

We provide two illustrative examples to explain these indirect interactions and their importance in our subsequent inference and network dynamics. The first one is an example of third-order interactions among the final states of a selected group of three lines which are undistinguished by direct observing pairwise correlations. The second example shows that we can mitigate the cascade effect by intentionally shutting down a line to exploit the possible negative interaction between a specific line group. We use data collected for failure cascading in power networks in data set D2.

Let i, j, and k denote, receptively, lines (3,5), (7,12), and (5,6) in the IEEE-118 network[19]. We are interested in analyzing the failure statistics of these lines at the final states of
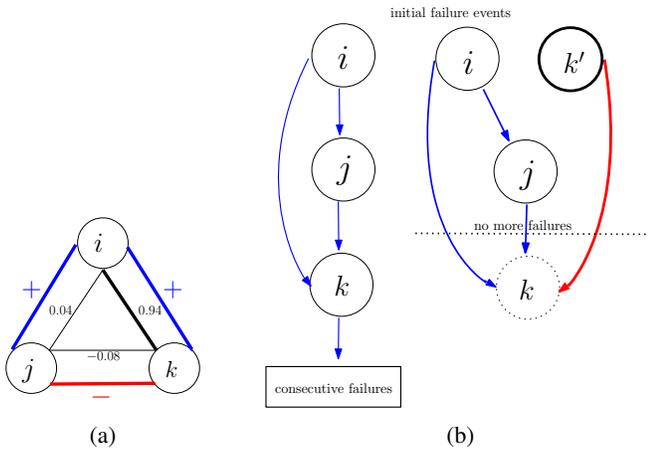
FIG. 1: (a) The three-way interactions among three selected lines are shown as a frustrated triplet. The pairwise Pearson correlation coefficients are shown in the inner triangle. We show positive interactions in blue and negative interactions in red. Due to the negative interaction between k and j, we do not observe a significant correlation between the failures of i and j. (b) Compared with the initial failure of i (left), the intentional failure of k' after the initial failure of i (right) avoids subsequent failure of k and the following cascading due to negative interaction between the failure of k and k'.

cascade trajectories. Such analysis is helpful, for example, in finding co susceptible groups of lines in cascade scenarios.

Assume C_{xy} denote the Pearson correlation coefficient between x and y. Using data set D_2 we have $C_{ik} = 0.94$, $C_{ij} = 0.04$, and $C_{kj} = -0.08$. See Fig. 1(a). Therefore, pairwise correlations show that the failure of lines i and k are strongly correlated, and there is no significant correlation between the failures of i and j. Now let $C_{x,y|z}$ denote the correlation between lines x and y given the state of line z. We have $C_{i,j|k=-1} = 0.43$ and $C_{i,j|k=+1} = -0.005$ where in 23499 final state samples we have $k = -1$ and for 13575 final state samples $k = +1$. If line k does not fail, then there is a significant correlation between the final state of i and j, while if line k fails, there is not. Here, we observe statistically significant three-way interaction, which is undistinguished by observing pairwise correlations as the pairwise correlations do not distinguish different conditional interactions.

Next, let J_{xy} denote the interaction value for lines x and y predicted by the learned statistical models in Section IV. The learned model predicts strong positive bi-directional interaction between i and j and so do k and i, i.e., $J_{ij}, J_{ji} \gg 0$ and $J_{ki}, J_{ik} \gg 0$. However, it predicts strong negative bi-directional interaction between j and k, i.e., $J_{jk}, J_{kj} \ll 0$. We find that the weak correlation between the states of i and j roots in the strong negative interaction between the failure of j and k. In scenarios in which i and k fail, j did not fail, consistent with the data. That is in total 13575 final state samples with $k = 1$ in 13165 samples i and k fail and j did not fail. These third-order interactions, named the frustrated triplets, are not considered by simply looking at the pairwise correlations. This example

shows that we can not rely on the naive pairwise correlation coefficient, for example, to infer the groups of lines that fail together as some strong interaction might be undistinguished.

Fig. 1(b) shows another example of the importance of finding the higher-order interaction in the cascade dynamics. In this example we have $i = (26, 25), j = (30, 38), k = (17, 18)$ and $k' = (18, 19)$. Here we observe how the strong negative interaction between the failure of line k and k' can mitigate the cascade effect. The initial failure event of the line i leads to overload and failure of the line j. Next line k fails, and we observe a series of consecutive line failures that fails 12 other lines. However, if we intentionally fail k' after the initial failure of i, we observe that j fails and the process stops. Our temporal interaction analysis in Section V shows that there is a strong negative interaction between the failures of k and k'; suggesting that we can prevent the failure of k and its subsequent failures by intentionally failing k' in this scenario.

IV. STATIC INTERACTION LEARNING

This section is interested in finding the static interaction graph, i.e., the relationship between a pair of lines' states at the final state of cascades called steady states. Recall that no other failure happens in steady states, and all network constraints are met. We note that due to global (e.g., power balance at each island and maximum power capacity of generators) and high density of local constraints (e.g., flow conservation rule at each node and flow capacity of each line), the number of such steady states is limited and is much less than 2^L. The outcome of these constraints is that there exist specific network states that can be learned in terms of effective pairwise interactions in which one can learn the line i state, s_i, given the states of its influential neighbors ∂_i, $\mathbf{s}_{\partial_i}$. The static interaction graph helps us understand which links tend to fail together and find co-susceptible groups. Also, we can reconstruct these final states using Glauber dynamics as discussed in IV B.

Also, note that according to the nature of the power networks, the desired interaction matrix is not symmetric in general. Consider lines e and $\hat{e}$ which, respectively, connect a generator and a load to the network in a nearby neighborhood. The network is subject to tight constraints after e fails, which probably leads to $\hat{e}$ failure. The failure of $\hat{e}$, on the other hand, makes the constraints lose and provide more slack power for the network.

A. Logistic regression model

Let single out link i and assume that we have other links' states at time t denoted by $\mathbf{s}_{-i}(t)$. We can find $(h_i, \{J_{ij}, j \neq i\})$ such that the probability that link i at $t + 1$ is at proper state consistent with the data (constraints) is maximized where J_{ij} is the influence of line j on line i and h_i is a local factor. Specifically, let the state of link i be related to other links' states according to

$$\Pr(s_i(t+1)|\mathbf{s}_{-i}(t)) = \frac{1}{2}\left[1+s_i(t+1)\tanh(H_i(t))\right] \quad (1)$$

$$= \frac{1}{1+e^{-2s_i(t+1)H_i(t)}},$$

$$H_i(t) = h_i + \sum_{j\neq i} J_{ij}s_j(t).$$

Equ.(1) is a logistic regression estimator for s_i conditioned on other links' states as the most widely used multivariate nonlinear statistical model. We should find $(h_i, \mathbf{J}_i)$ by maximizing the log-likelihood function of observing M independent $s_i(t+1)$ given $\mathbf{s}_{-i}(t)$ over the data by $(h_i^*, \mathbf{J}_i^*) = \mathrm{argmax}_{(\mathbf{h}_i, \mathbf{J}_i)} \mathscr{L}_D(h_i, \mathbf{J}_i)$ where

$$\mathscr{L}_D(h_i, \mathbf{J}_i) = \frac{1}{M}\ln\prod_{m=1}^{M}\Pr(s_i(t+1)|\mathbf{s}_{-i}(t))$$

$$= \left\langle \ln\frac{1}{1+e^{-2s_i(t+1)(h_i+\sum_{j\neq i}J_{ij}s_j)}} \right\rangle_D. \quad (2)$$

$\mathbf{J}_i$ is the ith row of interaction matrix and $\langle f(\mathbf{s})\rangle_D = \frac{1}{M}\sum_{m=1}^{M}f(\mathbf{s}^{(m)})$ with data set $D = \{\mathbf{s}^1, \ldots, \mathbf{s}^M\}$.

In practice, however, link i does not effectively interact with all other links, and we are interested in finding a sparse solution in which the state of each link is presented in terms of explainable interactions that the physics of the problem dictates. In the l_1-regularized learning technique, to avoid finding spurious meaningless interactions, the penalizing term is added to the objective function of (2) considering the prior knowledge of the interactions. This penalizing term leads to set un-explainable interactions to zero.

Let ∂_i denote the neighbors of link i, i.e., the set of other lines with them i has effective interaction. In[22] the authors show that reconstruction of the interaction structure and strength is possible with a two-stage algorithm. In the first stage, we find the underlying graphical model by ruling out the weak interactions and finding the explanatory neighbor variables, $\partial_i, \forall i$. In this regard, we first solve L independent optimization problems as

$$(h_i^0, \mathbf{J}_i^0) = \mathrm{argmax}_{(\mathbf{h}_i, \mathbf{J}_i)}\mathscr{L}_D(h_i, \mathbf{J}_i) - \lambda\sum_{j\neq i}|d_{ij}J_{ij}|, \quad (3)$$

where λ is a regularization parameter and d_{ij} is the distance between line i and j according to definition in subsection III A. Here, we use the prior knowledge that the physically adjacent lines show greater interaction absolute value and hence less penalize the corresponding interaction in the optimization objective. Then all weak interactions with $-\delta_m < J_{ij} < \delta_p$ are set to zero, where $\delta_m, \delta_p > 0$ are proper thresholds.

In the second stage, having the interaction structure, we find the interaction strength $(h_i^*, \mathbf{J}_i^*)$ by solving (3) again with $\lambda = 0$. Note that we may end up with weak but important coupling at the end of the procedure.

Choosing the appropriate λ is related to the graphical model reconstruction problem and should be tuned for the inference problem. Assuming no other prior information, this

parameter is related to the number of samples M, number of variables, L, and the accepted error in interaction graph reconstruction ε, by $\lambda \propto \sqrt{\ln(L^2/\varepsilon)/M}$[22]. δ_p and δ_m are then selected by inspecting the histogram of $\mathbf{J}_i$ values near zero and identifying the gaps in the density of interaction strengths.

Note that by proper selection of $\lambda, \delta_p, \delta_m$ we can trade off the goodness of fit to data for the model complexity or finding a sparse interaction matrix. Also, the l_1-regularized logistic regression in (3), is the conditional maximum entropy inference of $s_i(t+1)$ given $\mathbf{s}_i(t)$, and benefits from the learning guarantees of this model[23].

Computing the derivative of $\mathscr{L}_D(h_i, \mathbf{J}_i)$ with respect to h_i and J_{ij}, at the optimal point, we have

$$\langle s_i\rangle_D \approx \left\langle \tanh(h_i^* + \sum_{k\in\partial_i}J_{ik}^*s_k)\right\rangle_D \quad (4)$$

$$\langle s_is_j\rangle_D \approx \left\langle s_j\tanh(h_i^* + \sum_{k\in\partial_i}J_{ik}^*s_k)\right\rangle_D$$

which can be used as a measure of goodness of fit.

Learning $(\mathbf{h}^*, \mathbf{J}^*)$, we can use a dynamics which updates one link (spin) at each time step according to (1) to find steady states. The Glauber dynamics is widely used in statistical physics for describing a single site dynamics and finding the equilibrium and non-equilibrium Ising models. The Glauber process starts with a random initial site configuration. Next, at each time step one site is selected randomly, say i, and updated, i.e., $s_i(t+1)$ takes value one with probability $\Pr(s_i(t+1) = 1|\mathbf{s}_{-i}(t)) = \frac{1}{1+e^{-2\left(h_i+\sum_{j\in\partial_i}J_{ij}s_j(t)\right)}}$. In the next subsection, we show that the Glauber dynamics can successfully reconstruct the network steady states using the learned interaction matrix.

B.   Interactions at steady states

Since multiple initial failures may lead to the same steady state we first remove the final duplicate states in each data set. In the learning procedure, we use $\lambda_1 = 0.0001$ and $\lambda_2 = 0.0005$ for data sets $D1$ and $D2$. Also, we set $\delta_m = \delta_p = 0.1$ to learn $(\mathbf{h}_i, \mathbf{J}_i)$ for all i. The maximum edge distance for the IEEE-118 network is 15.

The optimization problem in (3) is convex and hence has a unique global optimum. However, the objective function is not differentiable if $\lambda \neq 0$. Therefore, in the first stage of the algorithm, we use proximal gradient descent, which shrinks the non-explanatory variable to zero in the projection step to find $(h_i^0, \mathbf{J}_i^0)$ for each i.

Using the selected parameters, we find sparse interaction matrices. The ratios of non-zero elements in $\mathbf{J}_1^*$ and $\mathbf{J}_2^*$ to all possible $L(L-1)$ interactions are 6.5% and 5.8%.

Figs. 2a and Figs. 2c show goodness of fit for the estimated $\langle s_i\rangle$ and $\langle s_is_j\rangle$ reconstructed from Equ.(4) against the values computed from the corresponding data set where $\langle s_i\rangle = \langle s_ir_0\rangle$ with $r_0 = 1$. The figures show that the learned models fits to the corresponding data. Also, we notice that using data set

$D1$ we observe only positive $\langle s_i s_j \rangle$ for pairwise interactions. However, in data set D_2, we have pairs of links with $\langle s_i s_j \rangle \leq 0$ which means we have lines i and j with $s_i = -s_j$, i.e., only one of them fails in steady state. This observation is the effect of indirect interactions in severe cascading scenarios which is not observed in the normal operation of a power system. Its physical meaning shows that the network partitions in cascading scenarios.

We can consider the power network as a system that works in a steady state and is subject to random perturbations due to line failures. The initial failure may cascade until the network settles down in a new steady state. Using the learned interactions, we can update the states of other lines using Glauber dynamics. The Glauber dynamics uses the learned interactions to update the state of a single line considering the states of others until it settles down in a steady state in which it remains for a long time. Therefore, we should observe the steady states much more than transient ones in a long run of updates. We use Glauber dynamics to sample these steady states and compare the statistics of these samples against the data to show the predictive performance of the learned interaction matrix. We consider the statistical steady states of the Glauber dynamics as the final states of cascade samples.

We generate M samples using the Glauber dynamics starting from a random initial $\mathbf{s}(0)$ in which each state sets uniformly $+1$ or -1 with probability 0.5. Therefore, the Glauber dynamics starts with an initial state far from the network's steady states used in the training phase, and we need many updates in the Glauber dynamics.

We set the warm-up time to $10^3 L$ in Monte Carlo simulations and the Monte Carlo step to $20L$ between sampling. Fig. 2b and Fig. 2d, show the $\langle s_i \rangle$ and $\langle s_i s_j \rangle$ from these samples against the values in corresponding data sets. Our extensive numerical study shows that the reconstruction of weak (near zero) and negative $\langle s_i s_j \rangle$ from the Monte Carlo samples is very hard and corresponds to sampling rare events from a dynamical system. This observation also emphasizes that relying on just positive correlations between the line failure is insufficient to understand the system's behavior in large cascades.

To evaluate the predictive capability of the model, we next compare the complementary cumulative distribution function (CCDF) of cascade size, P_Z, for steady state configurations in the Monte Carlo (MC) samples against the data in Fig. 3a and Fig. 3c. The maximum cascade size, the maximum number of failed links, in the data sets are $Z^{max}_{D_1} = 84$ and $Z^{max}_{D_2} = 85$ and in the MC samples are $Z^{max}_{MC_1} = 66$ and $Z^{max}_{MC_2} = 79$. As expected, the model learned with more extreme samples better captures the link states in the cascading scenarios. The inset of the figures compares binned probability of the cascade size in which we plot $p_Z(z) = \Pr(z \leq Z \leq z + \Delta z)$ with $\Delta z = \frac{Z^{max}_D}{20}$ for the MC samples against the values in the corresponding data set. We note that the density function of cascade size, $p_Z(z)$ spans three orders of magnitude, indicating the power-law distribution at the tail. Also, the model successfully generates samples whose density function spans this range.

In another predictive experiment, we generate new 5000 failure trajectories independently of the training data sets and evaluate how the learned model predicts the state of a spe-
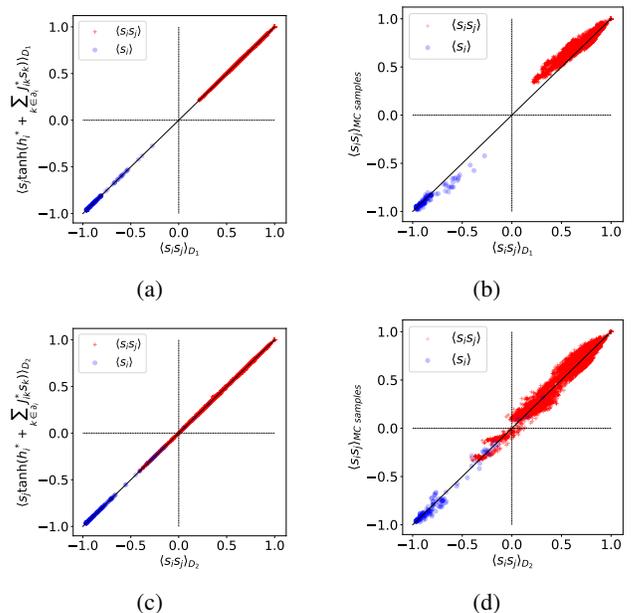
FIG. 2: Reconstructed $\langle s_i \rangle$ and $\langle s_i s_j \rangle$ against the actual values from data (a,c) reconstructed by applying the learned parameters on the data set D_1 and $D2$, and (b,d) using the Monte Carlo samples drawn from Glauber dynamics for data sets D_1 and D_2.

cific link given the others' states. For each new sample, we select a link with state $+1$ or -1 with the probability of 0.5. We then predict the selected link's true state probability using the model, assuming that the other links' states are available. Also, we perform the same experiment when we add perturbation to the given states by randomly selecting two neighboring links of the selected link and intentionally flipping their states. Fig. 3b and Fig. 3d show the corresponding Receiver Operating Characteristics (ROC) curves for data sets D_1 and D_2. The ROC curve shows the predictor's performance by depicting the true positive rate against the false positive rate for different thresholds. The models fairly predict the true failure probability of the selected links. The decrease in the ROC's AUC (area under the curve) with perturbations shows the model's sensitivity to perturbing explanatory variables.

C. Inference using interaction matrix

In this section, we use the static interaction matrix to infer some structural properties of the network. We study the regularities in the interaction graph, $\tilde{\mathscr{G}}$, which corresponds to the interaction matrix $\mathbf{J}$ to find links that fail together. $\tilde{\mathscr{G}}$ is a weighted, signed, and directed graph with L nodes in which a link $i \rightarrow j$ shows that line i affects the state of the line j.

We are interested in finding co-susceptible groups of lines that tend to fail together statistically. We use the Infomap[24] as an appropriate algorithm with proper weights for each interaction to find clusters of nodes with the same states in different network steady states. Infomap is a flow-based cluster-
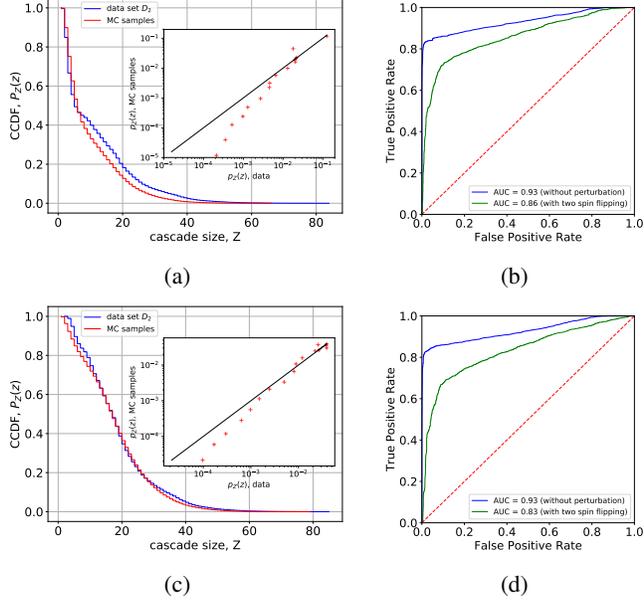
FIG. 3: (a,c) CCDF of the cascade size from the data sets and the MC samples, the inset compares the binned probability of the cascade size for the MC samples against the values in the corresponding data set. (b,d) The ROC for predicting the state of a selected link without and with two neighbor links state flipping.

ing mechanism that finds the organization based on the real flow of interactions in the underlying network. Here, we use Infomap to capture the desired failure propagation dynamics (flow) in our directed, and weighted interaction graph[25].

We first convert the interaction values to proper positive weights, which the random walker subsequently uses in the network as a proxy of failure flow in the network. Let $p_i = \Pr(s_i = +1 \,|\, \mathbf{s}_{\partial_i})$ where we remove time dependency for short writing. In the binary logistic regression learning we find $(h_i^*, \mathbf{J}_i^*)$ such that $\log \frac{p_i}{1-p_i} = 2(h_i^* + \sum_{j\in\partial i} J_{ij}^* s_j)$, i.e., we find the log-odds of line i failure in terms of the explanatory neighboring links' states. Now, assume the random walker is at node $j \in \partial_i$ of $\tilde{\mathscr{G}}$. The state of node j contributes in node i' state according to $[\mathbf{J}]_{ij}$. Let $p_{ij}^+ = \Pr(s_i = +1 \,|\, s_j = +1, \mathbf{s}_{\partial_i \setminus j})$ and $p_{ij}^- = \Pr(s_i = +1 \,|\, s_j = -1, \mathbf{s}_{\partial_i \setminus j})$. Using (1) we observe that[26]

$$e^{4J_{ij}} = \frac{p_{ij}^+(1-p_{ij}^-)}{p_{ij}^-(1-p_{ij}^+)}. \tag{5}$$

We can interpret p_{ij}^+ as the probability of failure flow from j to i for a given $\mathbf{s}_{\partial_i \setminus j}$ where $\frac{p_{ij}^+}{1-p_{ij}^+}$ is the corresponding odds. Correspondingly, p_{ij}^- is the probability of failure flow from i's neighbors except j to i. The ratio $\left[p_{ij}^+/(1-p_{ij}^+)\right]/\left[p_{ij}^-/(1-p_{ij}^-)\right]$ is a good measure for the share of failure flow from j to i. Therefore, we assign $e^{4J_{ij}}$ as the weight of link $j \to i$ in $\tilde{\mathscr{G}}$.

If J_{ij} is sufficiently positive, then $p_{ij}^+ \gg p_{ij}^-$ and if J_{ij} is sufficiently negative $p_{ij}^+ \ll p_{ij}^-$. Note that weak coupling $J_{ij} \approx 0$ means $p_{ij}^+ = p_{ij}^-$ and as expected does not contribute much in clustering process. We run the two-level Infomap clustering algorithm on data sets D_1 and D_2 and sort the clusters based on their sizes. The nodes of $\tilde{\mathscr{G}}$ (lines of $\mathscr{G}$) belong to the same cluster, then get sequential indices.

Fig. 4 shows the results for both data sets where we sort clusters according to their sizes and assign consecutive indices to lines in the same clusters. Infomap finds 8 and 15 clusters with cluster size greater than two for D_1 and D_2. The models suggest that there exists a clustering structure in the line failure in both data sets. As expected, the nearby lines are mostly in the same cluster. We, however, observe distant lines which are grouped in the same cluster. Furthermore, the clustering result for data set D_2 shows more distinctive clusters roots to line pairs with $\langle s_i s_j \rangle \approx 0$.

Let random variable $Z_{\mathscr{C}} = \sum_{j \in \mathscr{C}} \frac{(1+s_j)}{2}$ denote the number of failures in a final steady-state cascading trajectory for cluster $\mathscr{C}$. We compute $\Pr(Z_{\mathscr{C}} = z \,|\, Z_{\mathscr{C}} > 0)$ by marginalizing over the other lines' states in the data set to find to what extent the failure of one line in the group leads to other lines' failures in this group. The null hypothesis is to select a subset of lines randomly and uniformly, $\mathscr{R}$, with the same cardinality, i.e., $|\mathscr{C}| = |\mathscr{R}|$, and compute the same measure. The ratio of $\gamma = \frac{\mathbf{E}\left[Z_{\mathscr{C}} = z \,|\, Z_{\mathscr{C}} > 0\right]}{\mathbf{E}\left[Z_{\mathscr{R}} = z \,|\, Z_{\mathscr{R}} > 0\right]}$ then shows the effectiveness of the clustering method against the null hypothesis. Here $\mathbf{E}$ denotes the expectation value of the desired co-failure measure. Fig. (4c) and Fig. (4d) show the distribution of the γ values for 200 random samples as a box plot chart for cluster sizes greater than four, where the triangle token shows the mean and the horizontal bar in each box is the median of samples. We observe that except for one cluster in data set D_2, the mean values of the co-susceptibility measure γ in the Infomap clusters are approximately one order of magnitude greater than the null hypothesis.

V. TIME SERIES INTERACTION MODELING

The objective of this section is to learn how the states of links change over time. Instead of updating a specific link state near the steady states, we find the interaction matrix that encodes how the cascade unfolds in the network. The importance of this problem is to design mitigation strategies for power networks.

Each trajectory in our data sets captures the sequence of all link states until the network settles in a steady state. Therefore, each trajectory is a time series of links' states $\mathbf{s}(0), \mathbf{s}(1), \ldots \mathbf{s}(t_{ss})$ where t_{ss} is the time that failure propagation ends. The next state of the final steady state is itself, $\mathbf{s}(t_{ss} + 1) = \mathbf{s}(t_{ss})$ means that no more updates happen for this current state in the learning. For each data set we remove possible duplicate trajectories due to the same initial failure, and find $T = \sum_{j=1}^{M_0} 1 + t_{ss}^j$ consecutive network's state.

(a)   (b)

(c)   (d)
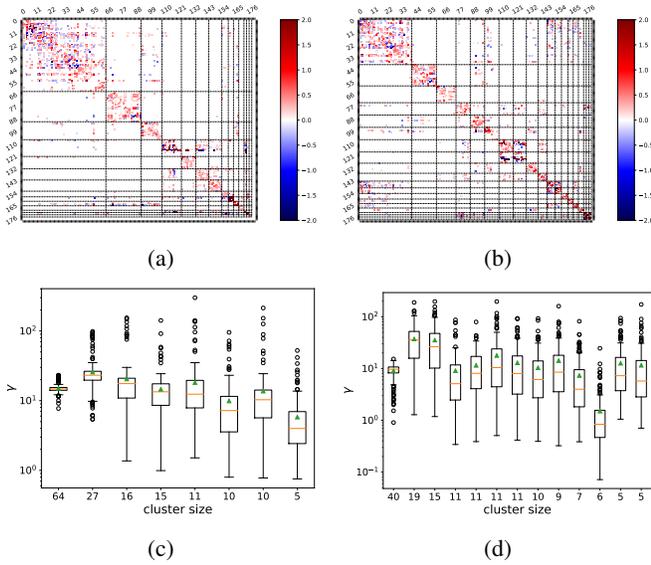
FIG. 4: (a,b)-heat map of the interaction matrix when lines are grouped and reindexed sequentially based on the Infomap clustering of the corresponding interaction graph $\mathscr{G}$ for (a) data set D_1 (b) data set D_2. the thin dashed lines separate different clusters. (c,d)-the box plot of the γ values where the triangle token shows the mean and the horizontal line in each box shows the median for (c) data set D_1 and (d) data set D_2

A. Logistic regression model

We adopt the kinetic Ising model with asynchronous updates[27]. In this model, at each time step the state of each link is updated with the probability given in (1) which can be read as $\Pr(s_i(t+1)|\mathbf{s}_{-i}(t)) = \frac{e^{s_i(t+1)H_i(t)}}{2\cosh H_i(t)}$. Note that the deployed model and data sets of steady states in the previous section can be considered as one step kinetic Ising model. The log likelihood function is

$$\mathscr{L}_D(\mathbf{h},\mathbf{J}) = \frac{1}{T}\sum_{t=1}^{T-1}\sum_{i=1}^{L}\left[s_i(t+1)H_i(t) - \ln 2\cosh(H_i(t))\right]. \tag{6}$$

The objective is finding $(\mathbf{h},\mathbf{J})$ which maximize the desired l_1-regularized function

$$(\mathbf{h}^*,\mathbf{J}^*) = \mathrm{argmax}_{(\mathbf{h},\mathbf{J})}\mathscr{L}_D(\mathbf{h},\mathbf{J}) - \lambda\sum_{j\neq i}|d_{ij}J_{ij}|, \tag{7}$$

In contrast to the previous section in which we solve an optimization problem for each link independently, we should find $(\mathbf{h},\mathbf{J})$ in an optimization problem over L^2 variables. Likewise, we follow a two-stage algorithm in the previous section to find the most explanatory interactions and fine-tune them. Since these are convex optimization problems, there are very efficient numerical methods to solve these problems. We use the naive gradient descent method to find the solution.

Computing the derivative of the likelihood function we have:

(a)   (b)
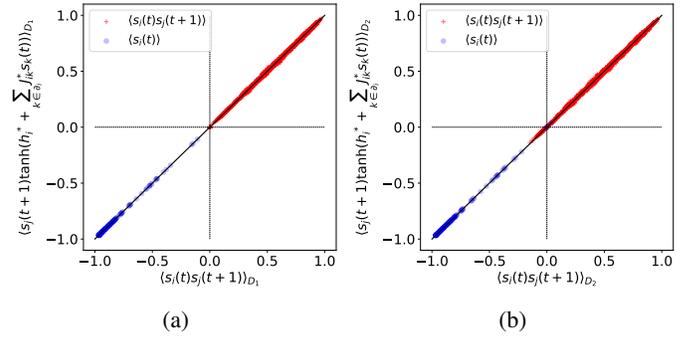
FIG. 5: Reconstructed $\langle s_i(t)\rangle$ and $\langle s_i(t)s_j(t+1)\rangle$ against the actual values from data sets for (a) data set D_1 (b) data set D_2.

$$\frac{\partial\mathscr{L}}{\partial h_i} = \frac{1}{T}\sum_{t=0}^{T-1}\left[s_i(t+1) - \tanh(H_i(t))\right] \tag{8}$$

$$\frac{\partial\mathscr{L}}{\partial J_{ij}} = \frac{1}{T}\sum_{t=0}^{T-1}s_j(t)\left[s_i(t+1) - \tanh(H_i(t))\right]. \tag{9}$$

Therefore, at the optimal point we have $\langle s_i(t)\rangle_D^t = \langle\tanh(H_i(t))\rangle_D^t$ and $\langle s_i(t)s_j(t+1)\rangle_D^t = \langle s_j(t)\tanh(H_i(t))\rangle_D^t$ where $\langle f(s(t))\rangle_D^t = \frac{1}{T}\sum_{t=0}^{T-1}f(s(t))$ which used as a goodness of fit measure.

B. Time series interactions

We set the same parameter values for λ and δ as the steady state analysis in order to find the corresponding $(\mathbf{h},\mathbf{J})$ for each data set. Fig. 5 shows that the model appropriately reconstructs $\langle s_i(t)\rangle$ and $\langle s_i(t)s_j(t+1)\rangle$.

The next step is to measure how the learned model predicts the failure unfolding in time. Here, we should select a threshold for binary decision-making at each step based on each line's predicted probability. We update the network state at each step and find consecutive network states in the time horizon. Note that the possible prediction error at the current time step will propagate to the consecutive time step predictions. Also, one should select the proper threshold for each line to improve the overall predictions. Here, we select the same threshold for all lines and use the model to predict the network state for the time horizon equals the corresponding trajectory's actual steps before settlement.

We find the consecutive network states for different threshold values and compare the predicted set of failed lines against the ground truth for 1000 independent trajectories of failure cascading. We compute the corresponding true positive and false-positive rates and find the ROC curve. Here false positive is predicting a line failure against the ground truth. See Fig. 6a. We repeat this experiment over another 1000 trajectories that last at least six-time steps to see how well the consecutive line failure prediction works—the corresponding ROC
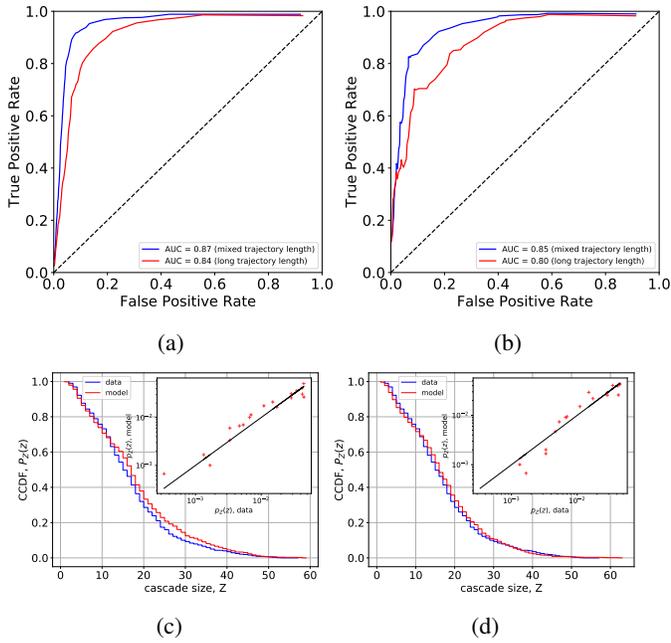
FIG. 6: (a,b)-the ROC curves for predicting the network state from the data and the model in time horizon compared to the ground truth trajectory for data set D_2. (c,d)-CCDF of the cascade size where the insets compare the binned probability of the cascade size for the model against the values of data. In (a,c), the time horizon is equal to the actual trajectory, and in (b,d) until no new updates happen in the network's state.

curve named as long-trajectories in Fig. 6a. We find similar results for data set D_1.

We provide the corresponding CCDF of cascade size, P_Z, for the model final state against the data in Fig. 6c. As in previous section, the inset of Fig. 6c compares binned probability of the cascade size in which we plot $p_Z(z) = \Pr(z \leq Z \leq z + \Delta z)$ with $\Delta z = \frac{Z_D^{max}}{20}$ for the model against the values in the corresponding data set which spans three orders of magnitude.

Finally, we repeat these experiments in the time horizon until no update happens in the network's state. Fig. 6b and Fig. 6d shows the corresponding ROC curve and the CCDF. These results show that the learned dynamic interaction matrix successfully predicts the network's state in consecutive time steps until settlement at the final steady-state.

VI. CONCLUSION AND FUTURE WORKS

Data-driven machine learning techniques can help better understand the complex dynamics of failure cascading in power networks, which involves higher-order interaction. We use regularized logistic regression-based machine learning tools to learn statistical models that capture pairwise and higher-order interactions of line failure cascading caused by line overloads and islanding in the linearized DC power flow model in power networks. The static model captures line failure interactions at the network's steady states by maximiz-

ing the likelihood of observing each line state given the other lines' states. We use this model to reconstruct the network's steady states, infer the cascade size statistics, and find co-susceptible line groups that fail together. The dynamic model learns the lines' states over the trajectories of failure cascading and can successfully predict the failure unfolding in the network. The results show that the machine learning approach and inference can help predict the final states of cascading and failure unfolding in power networks. The approach has the potential to be applied to other networked systems that might encounter similar failure phenomena. We will use these models to design cascade mitigation mechanisms in future works.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

[1] E. F. Moore and C. E. Shannon, "Reliable circuits using less reliable relays," Journal of the Franklin Institute **262**, 191–208 (1956).

[2] H. Ronellenfitsch, J. Dunkel, and M. Wilczek, "Optimal noise-canceling networks," Physical Review Letters **121**, 208301 (2018).

[3] B. A. Carreras, D. E. Newman, I. Dobson, and A. B. Poole, "Evidence for self-organized criticality in a time series of electric power system blackouts," IEEE Transactions on Circuits and Systems I: Regular Papers **51**, 1733–1740 (2004).

[4] B. A. Carreras, V. E. Lynch, I. Dobson, and D. E. Newman, "Complex dynamics of blackouts in power transmission systems," Chaos: An Interdisciplinary Journal of Nonlinear Science **14**, 643–652 (2004).

[5] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization," Chaos: An Interdisciplinary Journal of Nonlinear Science **17**, 026103 (2007).

[6] T. Nesti, F. Sloothaak, and B. Zwart, "Emergence of scale-free blackout sizes in power grids," Physical Review Letters **125**, 058301 (2020).

[7] L. Guo, C. Liang, and S. H. Low, "Monotonicity properties and spectral characterization of power redistribution in cascading failures," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (IEEE, 2017) pp. 918–925.

[8] Y. Yang, T. Nishikawa, and A. E. Motter, "Vulnerability and cosusceptibility determine the size of network cascades," Physical Review Letters **118**, 048301 (2017).

[9] P. D. Hines, I. Dobson, and P. Rezaei, "Cascading power outages propagate locally in an influence graph that is not the actual grid topology," IEEE Transactions on Power Systems **32**, 958–967 (2016).

[10] J. Qi, "Utility outage data driven interaction networks for cascading failure analysis and mitigation," IEEE Transactions on Power Systems **36**, 1409–1418 (2020).

[11] K. Zhou, I. Dobson, Z. Wang, A. Roitershtein, and A. P. Ghosh, "A markovian influence graph formed from utility line outage data to mitigate large cascades," IEEE Transactions on Power Systems **35**, 3224–3235 (2020).

[12] U. Nakarmi, M. Rahnamay Naeini, M. J. Hossain, and M. A. Hasnat, "Interaction graphs for cascading failure analysis in power grids: A survey," Energies **13**, 2219 (2020).

[13] F. Battiston, E. Amico, A. Barrat, G. Bianconi, G. Ferraz de Arruda, B. Franceschiello, I. Iacopini, S. Kéfi, V. Latora, Y. Moreno, *et al.*, "The physics of higher-order interactions in complex systems," Nature Physics **17**, 1093–1098 (2021).

[14] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek, "Weak pairwise correlations imply strongly correlated network states in a neural population," Nature **440**, 1007–1012 (2006).

[15] E. Aurell and M. Ekeberg, "Inverse ising inference using all the data," Physical Review Letters **108**, 090201 (2012).

[16] P. Crucitti, V. Latora, and M. Marchiori, "Model for cascading failures in complex networks," Physical Review E **69**, 045104 (2004).

[17] B. A. Carreras, V. E. Lynch, I. Dobson, and D. E. Newman, "Critical points and transitions in an electric power transmission model for cascading failure blackouts," Chaos: An interdisciplinary journal of nonlinear science **12**, 985–994 (2002).

[18] P. Hines, E. Cotilla-Sanchez, and S. Blumsack, "Topological models and critical slowing down: Two approaches to power system blackout risk analysis," in *2011 44th Hawaii International Conference on System Sciences* (IEEE, 2011) pp. 1–10.

[19] "Ieee 118 bus test case," `https://matpower.org/docs/ref/matpower5.0/case118.html`, accessed: 2021-11-190.

[20] I. Dobson, B. A. Carreras, D. E. Newman, and J. M. Reynolds-Barredo, "Obtaining statistics of cascading line outages spreading in an electric transmission network from standard utility data," IEEE Transactions on Power Systems **31**, 4831–4841 (2016).

[21] D. Witthaut and M. Timme, "Nonlocal effects and countermeasures in cascading failures," Physical Review E **92**, 032809 (2015).

[22] A. Y. Lokhov, M. Vuffray, S. Misra, and M. Chertkov, "Optimal structure and parameter learning of ising models," Science advances **4**, e1700791 (2018).

[23] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning* (MIT press, 2018).

[24] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," Proceedings of the National Academy of Sciences **105**, 1118–1123 (2008).

[25] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," The European Physical Journal Special Topics **178**, 13–23 (2009).

[26] G. Bresler, D. Gamarnik, and D. Shah, "Learning graphical models from the glauber dynamics," IEEE Transactions on Information Theory **64**, 4072–4080 (2017).

[27] H.-L. Zeng, M. Alava, E. Aurell, J. Hertz, and Y. Roudi, "Maximum likelihood reconstruction for ising models with asynchronous updates," Physical Review Letters **110**, 210601 (2013).