Energy Informatics

RESEARCH **Open Access**

Interaction graph learning of line cascading failure in power networks and its statistical properties

Abdorasoul Ghasemi[1,2*], Hermann de Meer[2] and Holger Kantz[3]

From The 12th DACH+ Conference on Energy Informatics 2023
Vienna, Austria. 4-6 October 2023. https://www.energy-informatics2023.org/

*Correspondence:
arghasemi@kntu.ac.ir

[1] Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran
[2] Faculty of Computer Science and Mathematics, Passau University, Passau, Germany
[3] Max Planck Institute for the Physics of Complex Systems, Dresden, Germany

Abstract

We consider line failure cascading in power networks where an initial random failure of a few lines leads to consecutive other line overloads and failures before the system settles in a steady state. Such cascades are rooted in non-obvious, long-range, and higher-order couplings among the lines' flows induced by physical constraints on the network. Failure interaction graph encodes which and to what extent other lines in a networked system are affected after each line failure and can help to predict the final state after an initial disturbance. We perform data analytics on the final lines' steady states of cascade trajectories to infer a specific line's state given the states of others. We use a generative model to reconstruct possible steady states, and a predictive model aims to predict the probability of each line's failures after the initial failure as a regression problem. The generative model uses regularized pseudolikelihood estimator to infer interaction weights by solving the inverse Ising problem and deploys Glauber dynamics to generate steady states. The discriminative model uses boosted trees to efficiently learn over training and predict over test data the state of each line as a target finding an appropriate subset of other lines' states as explanatory variables. We analyze the degree distribution of the corresponding interaction graphs to study the number of other components affected by each line failure (out-degree) or the number of lines that affect the state of a given line (in-degree). Both models show that the in-degree follows a power-law distribution. Finally, we discuss the possible application of the interaction graph for early link removal to mitigate the failure-cascading consequences.

Keywords: Cascading failure, Data analysis, Interaction graph

Introduction

Cascading failure is a complex process in distributed networked systems like power networks. It refers to a dynamic in which an initial, usually unforeseen disturbance, like the failure of a subset of components, leads to a sequence of dependent failures propagating through the network and causes severe damage (Dobson 2021). Although the probability of such failures is very low, the corresponding incurred direct and indirect cost is proportionally increased such that these failures are considered high-risk events. Indeed, the recorded data of blackouts in power grids (Carreras et al. 2004; Information on Blackouts in North America 2023) shows that the frequency and size of cascades because of unforeseen events like overload, protection failures, and human error, do increase over time, despite the high engineering efforts which are put in to make the system more robust.

Cascade dynamics is hard to predict because of specific features of the process. Cascading is a self-amplifying process that successively weakens the power network (Baldick et al. 2008) with three distinct features against other contact processes used to model disease epidemics and diffusion (Motter and Yang 2017): non-additive response, non-local propagation, and disproportional impact. The non-additive response refers to the fact that other components' responses may reinforce the initial failure's effect. The non-local propagation property of cascades reflects that the initial event may affect some distant components without affecting the in-between ones. Finally, failure of different components disproportionally impacts the number of other affected components, e.g., in power grids, there are a small, vulnerable set of components whose failures lead to large-scale outages (Yang et al. 2017a). Other studies show correlations between link failures in power grid networks, which reveal co-susceptible groups of links that tend to fail together (Yang et al. 2017b) though it is unclear how to find these sets.

From the computational perspective, the component failure does not have monotonicity property in the physical domain, i.e., failure of a set of components may have a smaller effect than the failure of a subset of them (Guo et al. 2017; Mazauric et al. 2013), and the problem of finding the set of lines whose removal has the most impact is NP-hard with different metrics. The authors of Ghasemi and Kantz (2022) show that the failure cascading in power networks involves higher-order interactions when the simultaneous states of a group of more than two lines affect the process dynamics. The authors observe positive and negative interactions between line failures and the existence of higher-order interactions like strongly frustrated triplets discussing non-monotonicity property and how the process could follow completely different trajectories considering the current state of two components making the trajectory prediction of failure cascading hard. In fact, a single error in the prediction procedure can completely change the successive predictions.

The large blackout failure cascades are rooted in diverse and complex interactions of components failure, protection device responses, and human intervention (Baldick et al. 2008) or interaction with the communication layer (Ghasemi and de Meer 2023) at different space and time scales, which might lead to rare, unpredictable outcomes. A failure interaction graph is an abstract graph different from the network's physical topology, which captures these interactions and could help in cascade failure analysis and mitigation. Here, we consider the problem of finding the lines' failure interaction graph that encodes which lines(and to what extent) are affected after a failure of a specific line.

We note that inferring the interaction graph is challenging because experimental studies on infrastructure networks are not practically possible, the available recorded data is rare and protected for security reasons, and various exogenous and endogenous disturbances affect the system. Therefore, modeling, collecting simulated data under controlled initial disturbances, and data analytics are crucial to learning explainable interactions.

Nakarmi et al. review techniques based on the physics of electricity and data-driven approaches to capture and infer the interactions in Nakarmi et al. (2020). In particular, correlation-based methods use pairwise statistical correlation without considering the order of failures. Furthermore, because of higher-order interactions, the naive correlations conceal some highly conditional correlated pairwise statistics as explained in Ghasemi and Kantz (2022). The consecutive pairwise methods consider the order of failures but ignore the impact of group interactions. The generation-based methods (Qi et al. 2014; Qi 2020) consider interactions between groups of lines' failures defining the generation concept; however, it is unclear how to separate different generations and consider pairwise interaction in consecutive generations to avoid underestimating/overestimating interactions.

In this paper, we argue that instead of learning the interactions using cascade trajectories, one can learn the interactions from the rather limited final states, which meet all system constraints as the trajectories attractors. As mentioned before, two different trajectories may differ slightly at the early stages of the cascade, making separating the generations of real data and model learning difficult. Furthermore, the failure trajectories are sensitive to details of the data collection simulation where it is unclear how much and which details should be considered in generating simulation data to capture the effective interactions. See Guo et al. (2017) for a review of failure-cascading simulation modeling in power networks and their advantages and disadvantages. However, the network's steady states depend only on the initial disturbance, even though it does not consider the order of failures. Data analysis and interaction learning using the steady states is more straightforward, requires much fewer data, and can still provide insights into the cascade process.

We simulate and collect cascade failure data on the IEEE-118 bus transmission network and perform data analytics on the steady states in section System model and data collection. Data visualization and analytics provides visualization of the steady states and the existence and importance of higher-order interactions among the network's steady states. In Learning interactions, we discuss two machine learning techniques to find the underlying interaction graph to robustly capture each line's state as a function of a subset of lines' states and analyze its statistical properties. The first model is based on a regularized pseudolikelihood estimator for solving the inverse Ising problem. Here, we use the learned interactions to reconstruct the steady states using the Glauber dynamics. The second model uses boosted trees as a highly predictive machine learning model where we use feature permutation to find the robust influential lines whose states affect the considered line's state. Our objective in both models is not to learn a highly predictive model by tuning the hyperparameters for specific test data but to find the underlying interaction graph's robust statistical properties and structure. In Statistical properties and using the interaction graph, we discuss that the corresponding interaction graphs show power-law in-degree distribution indicating that there exist lines in the network whose states are affected after any initial random failure. We also discuss using the interaction graph for early link removal to mitigate the cascade effects under specific scenarios before concluding the paper in section Conclusion and future works.

System model and data collection

We simulate, collect, and use a data set containing 50,000 trajectories of failure cascading on the IEEE-118-Bus transmission network with $N = 118$ buses and $L = 179$ links, in which all initial failures propagate at least one step. Some trajectories end at the same steady state where we have $M \approx 37000$ unique steady state. The network data (IEEE 2023) provides the topology, lines susceptances, demands, and generations at all buses as well as the maximum capacity for each line and the generation capacity of each generator.

We use the linear flow distribution (DC flow) model to find the lines' flows considering the generator with the greatest capacity as the slack bus generator. The initial disturbance is imposed by removing a small random subset of lines in which each line is removed independently with probability $p_f = 2.5/L$, i.e., in each initial failure, we remove 2.5 lines on average. The initial failure may lead to consecutive secondary line failures if a line's flow exceeds its capacity after flow redistribution. The flow redistribution is performed until no more failures happen, and the network settles into a steady state where all system constraints are met. The trajectories of failure unfolding and the corresponding final state are recorded. The flow computation is performed for each component separately if the network decomposes into components. See Ghasemi and Kantz (2022) for more details.

Let $s_i \in \{-1, +1\}$ denote the state of line i where $s_i = +1$ indicates that i initially or eventually fails. A network steady-state $\mathbf{s} = (s_1, \ldots, s_L)$ shows the state of each line at the end of a cascade trajectory and $Z = \sum_{i=1}^{L} (1 + s_i)/2$ denote the corresponding cascade size in terms of the number of line failures. As we will discuss, the cascade size shows a power-law distribution where we observe there exist initial random failures which cause nearly half of the lines to a failure state. We use $\langle f(\mathbf{s}) \rangle_D = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{s}^{(m)})$ to denote the empirical mean of $f(\mathbf{s})$ with data set $D = \{\mathbf{s}^1, \ldots, \mathbf{s}^M\}$.

Data visualization and analytics

This section provides data visualization of the cascade data and data analytics to emphasize the existence and importance of higher-order interactions in the lines' states.

Data visualization

Power networks are subjected to many constraints, including a high density of local constraints, e.g., flow conservation at each node and flow capacity of each line, and global constraints, e.g., power balance at each component or maximum generation capacity of generators. Therefore, the number of steady states is limited and much less than possible 2^L states, i.e., not all possible links' states are plausible. The states of stub lines connecting small demands do not affect other lines' states, whereas the state of a line that connects a big generator to the rest of the network influences many other lines' states. Therefore, we intuitively expect clusters of steady states that meet all system constraints with an associated cascade size.

Figure 1a shows a two-dimensional visualization map of the L-dimensional M steady states using the t-distributed stochastic neighbor embedding (t-SNE) method (Van der Maaten and Hinton 2008). t-SNE maps each high-dimensional point to the

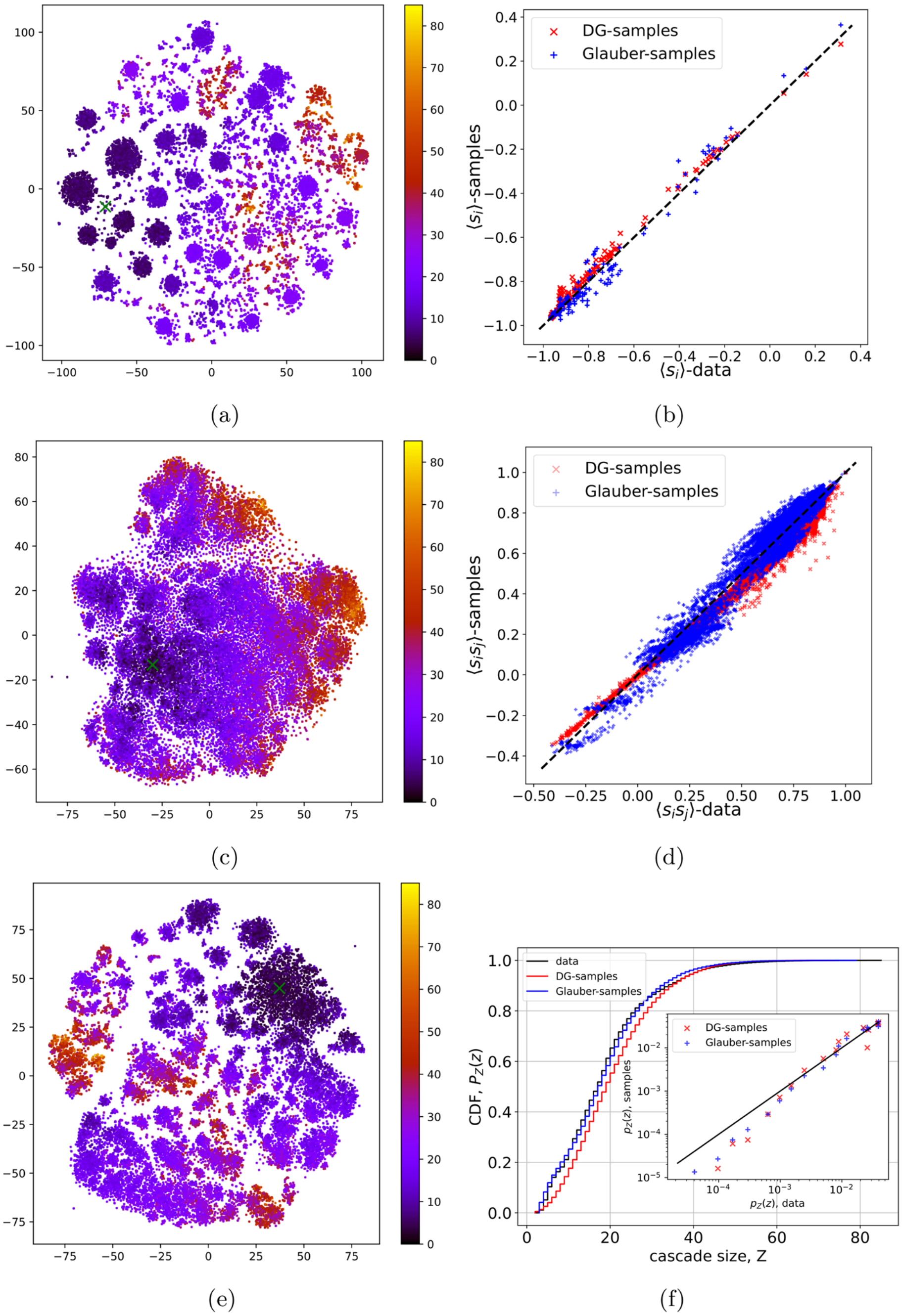


Fig. 1 2D visualization of the network's steady states for data, DG samples, and Glauber samples are shown in **a**, **c**, and **e**, where the color of each point shows the corresponding cascade size. The point depicted with the green cross symbol is the corresponding mapping of the network state when all lines work, $\mathbf{s} = (-1, \ldots, -1)$. The empirical mean and pairwise uncentered covariance of generated DG and Glauber samples' network states are approximately the same as data as depicted in **b** and **d**. Panel **f** shows the cascade size distribution for data and samples, and its inset depicts the binned probability of cascade size for samples against data

two-dimensional one by minimizing the divergence between the pairwise distances in the high and low dimensional space. We use the off-the-shelf t-SNE implementation to map all M steady states to a two-dimensional space adopting the Euclidean metric to measure the distance between data points. We should note that the exact distances between the two-dimensional space do not exactly reflect the corresponding distance in the L-dimensional space in the t-SNE algorithm, and the final map depends on the selected perplexity and learning rate parameters in the algorithm. Nevertheless, the whole map can provide insights into the data clusters.

We find that the number of possible network states that meet all system constraints is indeed limited, suggesting that one can infer the interactions by learning the statistical properties of steady states in these clusters. We observe the same clustering map of steady states with similar cascade sizes changing the perplexity and learning rate parameters. We learn models to predict the state of each line given the state of its influential neighbors in Learning interactions.

Importance of higher-order interactions

The outcome of the imposed constraints on lines' flows is diverse higher-order interactions among groups of lines at different spatial scales and sizes. Higher-order interactions (HOI) refer to an indirect interaction among a group of more than two lines where their simultaneous states affect the cascade dynamics. The failure correlation between two lines belonging to a higher-order group can not be distinguished by observing naive correlation coefficients but by conditioning on the states of other lines in the group. The authors of Ghasemi and Kantz (2022) show the existence of such HOI in power networks failure cascading.

To investigate the importance of HOI and the insufficiency of naive pairwise line statistics, we generate M distinct L-dimensional steady state samples with the same mean and covariance matrix as the data using the method based on dichotomized Gaussian (DG) distribution (Macke et al. 2009). This method first samples from L-dimensional Gaussian distribution with proper mean and covariance matrix and then maps each sample's element to 0 and 1 using the sign of the corresponding element in the continuous sample. The appropriate mean vector and covariance matrix are calculated based on the desired binary mean and covariance values. The entropy of the generated samples is close to the theoretical maximum entropy ensuring that the binary samples contain no more statistical information beyond the mean and pairwise statistics.

Figure 1b and d show that the generated samples from the DG distribution approximately have the same empirical mean, $\langle s_i \rangle$, and pairwise uncentered covariance, $\langle s_i s_j \rangle$, as data. Note that these samples are characterized by matching first and second moments with the data while exhibiting no additional discernible statistical regularities. Interestingly, the cascade size of DG samples has approximately the sample distribution as the data cascaded size, as shown in Fig. 1f. This result emphasizes that predicting the cascade size distribution is insufficient for evaluating the cascade models.

Fig. 1c shows a 2D visualization map of the DG-samples generated by the t-SNE method with the same parameters as Fig. 1a which does not show specific clustering structures. We perform this visualization with different perplexity values for t-SNE and found that the network state clustering in the data does not appear in DG samples. This visualization suggests that the pairwise statistics per se do not show the rich and complex interactions among the lines' states, indicating the importance of HOI. We will discuss the Glauber dynamics for generation samples in Learning interactions.

Learning interactions

This section explains two different methods for learning the interaction graph denoted by $\hat{G}$. The interaction graph is an abstract graph different from the network's physical topology, which reflects how one component's state affects others' components' states. $\hat{G}$ is a (weighted) directed graph in which each node corresponds to a physical line in the power network, and a link between j and i shows that the state of i is affected by the state of j in a defined manner. Therefore, the interaction graph encodes all influential neighbors of each line, which may not necessarily be adjacent in the physical network.

Interaction graphs can be constructed using prior knowledge and cascade data by applying machine learning (ML) tools. Designing ML tools for inferring interactions and using that for prediction and possible intervention is challenging. The reason is that the explainability of the learned model is as important as its prediction power. The model should meaningfully reveal possible higher-order or weakly positive/negative interactions. Furthermore, the amount of required data is unclear for many ML techniques, and we expect the model to predict unseen rare events. Nevertheless, as we will discuss, some persistence structures on the interaction graph exist, which are insightful for understanding the origins and mitigating the large cascades. Therefore, in the following, instead of focusing on tuning hyperparameters for improving the desired prediction power over test data, we focus on finding persistent structural features of interaction graphs among the applied ML techniques.

Regularized pseudolikelihood estimator

We use the regularized pseudolikelihood method for reconstructing the structure of interactions (Lokhov et al. 2018) by maximizing the probability of each line's state conditioned on the remaining ones and following the method suggested in Ghasemi and Kantz (2022). In this model, the state of the line i, s_i, is influenced by the states of other lines, $\mathbf{s}_{-i}$, in a probabilistic manner in terms of pairwise interactions given by $\Pr(s_i|\mathbf{s}_{-i}) = \frac{1}{1+e^{-2s_i\left(h_i+\sum_{j\neq i} J_{ij}s_j(t)\right)}}$. Here, h_i is a local factor reflecting the overload tolerance at the designing stage, e.g., the lines with higher maximum capacity tolerate more overload and are less susceptible to failure in the cascade process, and J_{ij} is the influence of line j on line i. The method first estimates an initial interaction strength of each line by maximizing a proper l_1- regularized pseudolikelihood over the M samples by solving

$$(h_i^0, \mathbf{J}_i^0) = \text{argmax}_{(\mathbf{h}_i,\mathbf{J}_i)} \left\langle \ln \frac{1}{1+e^{-2s_i(h_i+\sum_{j\neq i} J_{ij}s_j)}} \right\rangle_M - \lambda \sum_{j\neq i} |d_{ij}J_{ij}|, \tag{1}$$

where $\langle . \rangle_M$ denotes the average over M samples, and λ is a regularization parameter that should be tuned as a hyperparameter to push un-explainable interactions to zero. d_{ij} is the graph distance of lines i and j in the power network to impose a greater penalty for the potential influence of lines located at a greater physical distance to line i. Then, all weak interactions with $-\delta < J_{ij} < \delta$ are set to zero to find the influential neighbors of i, ∂_i where δ is a proper threshold. Finally, we find the interaction strengths by maximizing the pseudolikelihood over the selected neighbors without applying the regularization, $(h_i, \mathbf{J}_i) = \text{argmax}_{(\mathbf{h}_i,\mathbf{J}_i)} \left\langle \ln \frac{1}{1+e^{-2s_i(h_i+\sum_{j\in\partial i} J_{ij}s_j)}} \right\rangle_M$ (Lokhov et al. 2018). We examine different values for λ and δ to find parameters that better discriminate the clusters of meaningful and close to zero interactions in the strength histogram and select $\lambda = 0.0005$ and $\delta = 0.1$. See Ghasemi and Kantz (2022) for details.

Having found the effective neighbors and corresponding strength, we can use the model to generate the network's steady states. The idea is to consider the power network as a dynamic system that initially works in a steady state where all lines are working $s_i = -1, \forall i$ and $\Pr(s_i = 1|\mathbf{s}_{-i}) \approx 0$. The system is subject to random initial perturbations due to line failures that flip some random lines' states. This perturbation does change $\Pr(s_i = 1)$ for some i if the perturbed states intersect with the neighbor of i. By updating the state of all lines after the perturbation with their corresponding probability, we expect to push the network state to one of the steady states and remain at it until another perturbation applies. Therefore, in the long run of updates, we should observe the steady states much more than transient states, and if we sample from this dynamic, we expect to statistically capture samples corresponding to network steady states rather than transient states with less sojourn time.

We use sequential Glauber dynamics with the learned interactions to randomly select and update the state of each line, starting with a completely unplausible initial random state $\mathbf{s}_0$ in which each line is failed with probability 0.5. In each update, the algorithm selects a line uniformly and randomly, say i, and calculates the probability of updating its state to $+1$ using $p_i^+ = \frac{1}{1+e^{-2(h_i+\sum_{j\in\partial i} J_{ij}s_j)}}$. Next, it draws a random number from a uniform distribution in (0,1), say u, and updates the state of the line i using $s_i = \text{sign}\ (p_i^+ - u)$ where sign (.) denotes the sign operator. We draw M unique samples (name Glauber samples in figures) where the warm-up time is 10^3 updates, and the sampling step is $20L$ in Monte Carlo simulations.

Figure 1b and d show that the empirical mean and uncentered pairwise covariance of the Glauber samples are approximately the same as the data, and Fig. 1f shows that the corresponding cascade size distribution as well follows the data. The 2D visualization of these samples using the t-SNE method shows that the model can partially recover some network's state similar to the data shown in Fig. 1e, suggesting that the model learns some HOI beyond the first and second moments statistics.

The model's predictive power to predict a specific line's state, given the others, depends on properly selecting the model's hyperparameters λ and δ and selecting the proper threshold for making binary decisions following computing the selected line failure probability by the model.

We use the same model parameters as before, i.e., learned with $\lambda = 0.0005$ and $\delta = 0.1$, and evaluate its predictive power by computing the area under the precision-recall curve (AUC-PR). AUC-PR is a threshold-independent metric that considers the trade-off between the precision (the ratio of true-positive to the sum of true-positive and false-positive) and the recall (the ratio of true-positive to the sum of true-positive and false-negative). The AUC-PR metric is robust for highly imbalanced cascade data with rare binary events where the true negative (predicting a line does not fail consistently with the observed data) measure does not convey much information (Sofaer et al. 2019).

To generate test data, we use our simulator to generate 20,000 samples of the final network steady-state applying an initial random failure and find the AUC-PR measure for the lines that the percentage of observing $+1$ as target value is greater than 0.5 % total

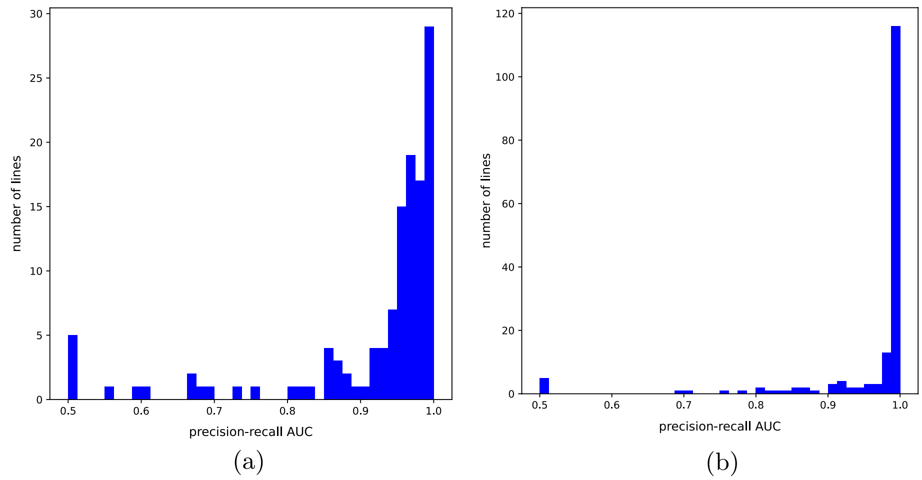


Fig. 2 Histogram of the number of lines with a specific area under the precision-recall curve for **a** Regularized pseudolikelihood estimator model and **b** the boosted tree model

number of target values. Figure 2a shows the histogram of the number of lines for different AUC values. We observe that for 109 out of the total 123 lines, the AUC measure is greater than 0.8. This result shows that the model has a reasonable prediction power for lines' states with a minimum positive final state in the data even though the hyperparameters did not tune to maximize any predictive power like AUC-PR over the dataset.

The big advantage of this generative model is that one can interpret the interaction weights. However, it is hard to learn the interaction for lines that do not have a minimum number of positive final states in the training data set.

XGboost regression

In this section, we use the scikit-learn XGBoost package as the off-the-shelf implementation of the well-known regularized extreme gradient boosting (XGboost) method as a powerful discriminative machine learning tool for regression problems (Chen et al. 2016). XGboost builds and uses an ensemble of decision trees to make predictions considering the trees' complexities in the regularization term. The method shows strong predictive power over different data sets but is not explainable like the Regularized pseudolikelihood estimator.

We train an xgboost regressor on the available training samples for each line, considering the selected line's state as the target and the other lines' states as explanatory variables with a binary logistic objective function. The model allows a l_1 regularization parameter, say α_i, to promote model sparsity and avoid overfitting in finding the best model for line i. Here, in contrast to the pseudolikelihood model, we tune the model of the line i by adjusting α_i using 5-fold cross-validation to maximize the AUC-PR on the training data.

We test the model prediction power by predicting the state of each line for the same test data as used for the regularized pseudolikelihood. Since xgboost is robust to imbalanced data, we use the model to predict the state of lines with at least one observed positive state in the test data. Fig. 2b shows the histogram of the number of lines for different values of AUC-PR, which shows that the model is indeed highly predictive.

Next, to select the robust influential neighbors of line i, we performed permutation importance for the state of the line $j \neq i$ as an explanatory variable over the test data. In permutation importance, the other lines' states are flipped one at a time, the model is re-fit, and the resulting decrease in model performance is measured to estimate the importance of each other line's state on the learned model. In order to get more reliable results, the permutation importance is repeated several times where the value of the selected line's state randomly changes, and the performance decrease is recorded. We then consider the median of each line's computed importances as a single measure of its importance. The median is a robust measure against outliers where we know that for each line, there do exist a few influential adjacent lines which strongly influence this line's state, as observed in the importance values histograms. Our objective is to separate those other lines which weakly influence the selected line from those with near-zero importance values. Therefore, all lines with an importance measure greater than 0.02 of the maximum importance value are selected as the influential lines to construct the interaction graph.

Statistical properties and using the interaction graph

Having the interaction graph with two distinct models, we can study its persistent statistical features. We are interested in the statistics of the number of lines that influence (in-degree) or are being influenced (out-degree) by each line's state. These statistics show the possible spatial scale that a line's state may interact and how much the system constraints are convoluted.

Degree distribution of interaction graph

Table 1 shows the basic statistics of the physical and interaction graphs. As expected, the average and standard deviation of interacting neighbors for a typical line is much greater than the physical network. Fig. 3 show the complementary cumulative distribution functions for the out- and in-degree distributions. We observed a power-law behavior at the tail of the in-degree distributions. The estimated exponent of the tail distribution is $\alpha_1 = 2.8$ for the first and $\alpha_2 = 3.1$ for the second model where $p(x) \propto x^{-\alpha}$ using (Alstott et al. 2014). This result suggests that there exist lines whose states are influenced by many others effectively after all initial random failures.

Table 1 The average and standard deviation of the physical network and the interaction graphs where graph1 refers to interactions inferred by the pseudolikelihood model, and graph2 refers to interactions inferred by boosted trees

	Avg. degree	Std. out-degree	Std. in-degree
Physical network	3.04	1.56	–
Interaction graph1	10.4	5.8	7.6
Interaction graph2	8.5	5.5	7.7

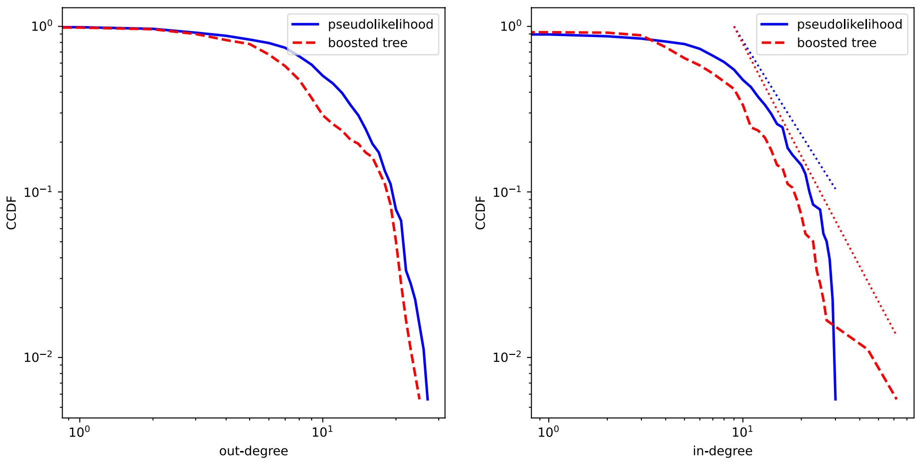


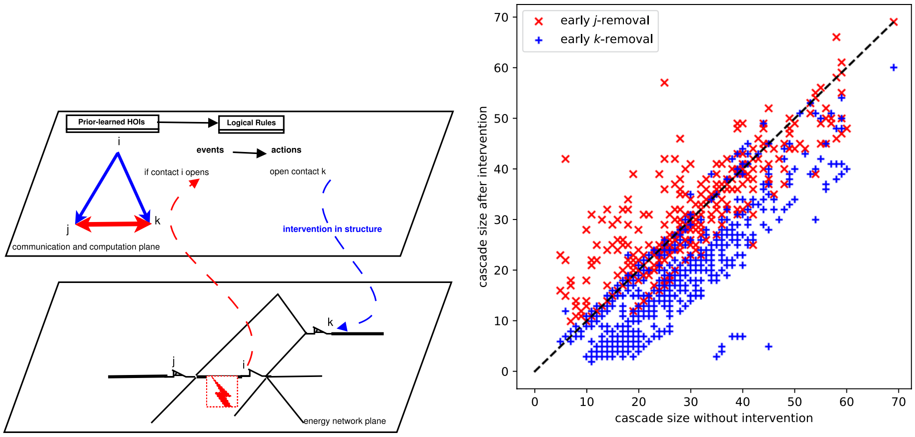
Fig. 3 The complementary cumulative distribution function for the out- and in-degree distributions of the interaction graph for the pseudolikelihood and boosted tree models. The dotted lines in the in-degree curve show the power-law fit at the tail of the distribution

Early link removal

The designed network structure at the planning stage is optimized for the normal operation and constraints of the system, considering the robustness (absorptive capacity) against a limited set of foreseen stresses. However, this structure may not be efficient or even imposes additional adversarial constraints after unforeseen stresses, here random line failure, which may amplify and propagate the initial failure impact. One approach is intervening and adapting the network structure to curtail the cascade impact. This intervention in network structure could be done by the intentional removal of a carefully chosen link(s) following the initial failure (Hines et al. 2009; Witthaut and Timme 2015). Nevertheless, it is unclear how to find these line(s) considering the combinatorial nature of the problem and the explosive number of possible initial failure scenarios and line(s) selection for removal.

Interaction graphs can be used for this intervention. We observe many negative interactions in the interaction graph inferred by the pseudolikelihood model. In particular, there exist strongly frustrated triplets. In a frustrated triplet, one line, say *i*, has positive interaction on the failure of the other two lines *k* and *j* when there is a negative interaction between these lines. Therefore, one expects that following the failure of *i*, either *j* or *k* may fail. Without any intervention, other lines' states determine which line may fail. However, one can avoid the consequences of *j* failure by early removing *k* or vice versa to steer the final network state to a more desirable steady state. Early link removal aims to intervene and proactively remove the specific line(s) after observing a particular transient network state.

Fig. 4a shows how to use the learned frustrated triplets at the computation plane to search for possible interventions in network structure. The objective is to use these HOIs to extract a table of logical rules for events and actions. Each event shows the current observable network state for a specific set of lines. The corresponding action determines which specific line's contact should be proactively opened to curtail the cascade.



(a) (b)

Fig. 4 (a)Intervention in network structure using learned triplets: Lines *i*, *j*, and *k* form a frustrated triplet according to the learned interactions. Failure of *i* has a positive interaction on *j* and *k*, and we observe a strongly negative interaction between *j* and *k*. After the failure of *i*, by intentionally early removal of *k* (*j*), one might avoid the failure of *j* (*k*) and the following consequences. (b) For a selected frustrated triplet, the data analysis shows that intentionally removing one of the links mitigates the cascade size in most random scenarios. Therefore, we add one entry to our logical rules: open the contact of *k* if the contact of *i* opens

We follow this idea with an illustrative example. By exploring the lines' interaction values of the pseudolikelihood model, we find that three lines $i = (85, 89)$, $j = (85, 88)$, $k = (82, 83)$ (see IEEE (2023) to locate these lines on the IEEE-118 bus network) form a strongly frustrated triplet. A close look at the cascade trajectories shows that from the total number of cases in which line i is initially disturbed, in near 94 %, line j and in 2 % k fails consequently. Also, we observe that the final cascade sizes in some instances are large.

Next, we consider the cascade size with early link removal of either j or k. We first separate all trajectories in which i and a random subset of other lines (except j and k) fail and find the corresponding cascade sizes. Then we find the cascade size for the same scenario when j or k is removed from the network after the same initial failure, i.e., j or k is added to the initial failure scenario, by running the simulator again for this intervened initial failure. Fig. 4b compares the cascade size results in these two scenarios. We observe that the early k-removal is beneficial in most cases even though it slightly increases the cascade size in some scenarios compared to non-intervention or proactively removing j.

We emphasize that using the learned models for intervention in the network using triplets needs that the model captures the physically meaningful interactions in the first place and should be further justified by other simulations. The former ensures that the model does not try to predict the current training/test data and reveals the real physical constraints of the network. The latter is because the lines typically belong to and are affected by different groups of higher-order interactions, not just the selected triplet.

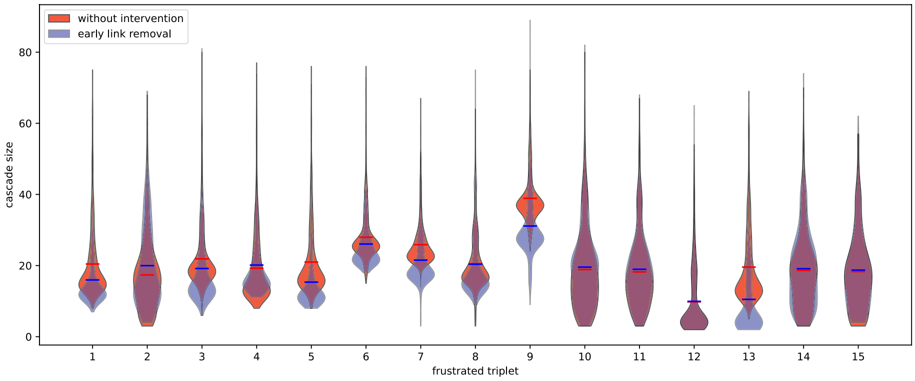


Fig. 5 The violin plot for each triplet shows the distribution of cascade size without and with proper early link removal

We extracted 15 strong frustrated triplets (those with strong positive and negative interactions) relying on the interactions learned by pseudolikelihood and performed the same experiment. We find the distribution cascade size with and without proper intervention and depict the results in Fig. 5. We observe that the early link removal can change the distribution of cascade size, and in 7 out of 15 triplets, by proper early link removal, one can decrease the average cascade size suggesting that interaction learning can be potentially used for cascade mitigation.

Conclusion and future works

In cascading failure dynamics in energy networked systems, the initial unforeseen failure leads to a chain of consecutive failures and may cause catastrophic outcomes. This dynamics roots in interactions among the system's components in different temporal and spatial scales with different sizes making the prediction of cascade behavior hard. We consider the cascade process after an initial random line failure which causes load redistribution and possible consecutive failures, and performed data analytics on the final states in which the network settles at them. Two models are learned to infer the final state of each line by finding the robust influential neighbors and inferring the interaction graph. The first model aimed to be interpretable, and the second model has more predictive power but is not easily interpretable. We discuss the possible application of interaction graphs for intervening in network structure, investigate their statistical properties, and show that their in-degrees follow a power-law distribution. Future works will consider how to infuse prior knowledge of the systems in learning interactions and find higher-order interactions to explain the cascade and mitigate its behavior.

About this supplement

This article has been published as part of Energy Informatics Volume 6 Supplement 1, 2023: Proceedings of the 12th DACH+ Conference on Energy Informatics 2023. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-6-supplement-1.

Author contributions

AG, HM, and HK designed the research. AG performed simulations and analyzed the empirical data. AG, HM, and HK wrote and reviewed the manuscript.

Funding
The work of A. Ghasemi was supported by the Alexander von Humboldt Foundation (Ref. 3.4-IRN-1214645-GF-E) for his research fellowship at the University of Passau in Germany.

Availability of data and materials
The datasets used and/or analyzed during this study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Published: 19 October 2023

References

Alstott J, Bullmore E, Plenz D (2014) powerlaw: a python package for analysis of heavy-tailed distributions. PLoS ONE 9(1):85777

Baldick, R., Chowdhury, B., Dobson, I., Dong, Z., Gou, B., Hawkins, D., Huang, H., Joung, M., Kirschen, D., Li, F., : Initial review of methods for cascading failure analysis in electric power transmission systems ieee pes cams task force on understanding, prediction, mitigation and restoration of cascading failures. In: 2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1–8 (2008). IEEE

Carreras BA, Newman DE, Dobson I, Poole AB (2004) Evidence for self-organized criticality in a time series of electric power system blackouts. IEEE Trans Circuits Syst I: Regular Papers 51(9):1733–1740

Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)

Dobson I (2021) Cascading network failure in power grid blackouts. In: Encyclopedia of Systems and Control, 199–202. Springer, Cham

Ghasemi A, Kantz H (2022) Higher-order interaction learning of line failure cascading in power networks. Chaos. 32(7):073101

Ghasemi A, de Meer H (2023) Robustness of interdependent power grid and communication networks to cascading failures. IEEE Trans Netw Sci Eng

Guo L, Liang C, Low SH (2017) Monotonicity properties and spectral characterization of power redistribution in cascading failures. ACM SIGMETRICS Perform Eval Rev 45(2):103–106

Guo H, Zheng C, Iu HH-C, Fernando T (2017) A critical review of cascading failure analysis and modeling of power system. Renew Sustain Energy Rev 80:9–22

Hines P, Balasubramaniam K, Sanchez EC (2009) Cascading failures in power grids. Ieee Potentials 28(5):24–30

IEEE 118 bus test case. https://matpower.org/docs/ref/matpower5.0/case118.html. Accessed 20 Apr 2023

Information on Blackouts in North America. https://www.nerc.com. Accessed 20 Apr 2023

Lokhov AY, Vuffray M, Misra S, Chertkov M (2018) Optimal structure and parameter learning of ising models. Sci Adv 4(3):1700791

Macke JH, Berens P, Ecker AS, Tolias AS, Bethge M (2009) Generating spike trains with specified correlation coefficients. Neural Comput 21(2):397–423

Mazauric D, Soltan S, Zussman G (2013) Computational analysis of cascading failures in power networks. ACM SIGMETRICS Perform Eval Rev 41(1):337–338

Motter AE, Yang Y (2017) The unfolding and control of network cascades. Physics Today 70(1):32–39

Nakarmi U, Rahnamay Naeini M, Hossain MJ, Hasnat MA (2020) Interaction graphs for cascading failure analysis in power grids: a survey. Energies 13(9):2219

Qi J (2020) Utility outage data driven interaction networks for cascading failure analysis and mitigation. IEEE Trans Power Syst 36(2):1409–1418

Qi J, Sun K, Mei S (2014) An interaction model for simulation and mitigation of cascading failures. IEEE Trans Power Syst 30(2):804–819

Sofaer HR, Hoeting JA, Jarnevich CS (2019) The area under the precision-recall curve as a performance metric for rare binary events. Methods Ecol Evol 10(4):565–577

Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res. 9(11)

Witthaut D, Timme M (2015) Nonlocal effects and countermeasures in cascading failures. Phys Rev E 92(3):032809

Yang Y, Nishikawa T, Motter AE (2017) Small vulnerable sets determine large network cascades in power grids. Science 358(6365):3184

Yang Y, Nishikawa T, Motter AE (2017) Vulnerability and cosusceptibility determine the size of network cascades. Phys Rev Lett 118(4):048301

Publisher's Note