Received: 28 September 2021 | Revised: 7 August 2021 | Accepted: 29 August 2021

IET Generation, Transmission & Distribution

DOI: 10.1049/gtd2.12287

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

A machine learning approach for real-time selection of preventive actions improving power network resilience

Matthias Noebels[1] | **Robin Preece**[1] | **Mathaios Panteli**[2]

[1] School of Electrical and Electronic Engineering, The University of Manchester, Manchester, UK

[2] Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

Correspondence
Mathaios Panteli, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus.
Email: panteli.mathaios@ucy.ac.cy

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Numbers: EP/L016141/1, EP/R030294/1; Newton Fund, Grant/Award Number: Project 1304

Abstract

Power outages due to cascading failures which are triggered by extreme weather pose an increasing risk to modern societies and draw attention to an emerging need for power network resilience. Machine learning (ML) is used for a real-time selection process on preventive actions, such as topology reconfiguration and islanding, aiming to reduce the risk of cascading failures. Training data is obtained from Monte Carlo simulations of cascading failures triggered by extreme events. The trained ML-based decision-making process uses only predictors that are readily available prior to an extreme event, such as event location and intensity, network topology and load, and requires no further time-consuming simulations.The proposed decision-making process is compared to time-consuming but ideal decision-making and fast but trivial decision-making. Demonstrations on the German transmission network show that the proposed ML-based selection process efficiently prevents the uncontrolled propagation of cascading failures and performs similarly to an ideal decision-making process whilst being computationally three orders of magnitude faster.

1 | INTRODUCTION

During the last decades, power outages due to cascading failures and triggered by extreme weather have become more frequent and severe [1, 2]. Extreme weather events, such as hurricanes or flooding, are likely to occur more often in the future because of climate change [3, 4]. The network is also put under additional stress due to many strategies that aim to reduce carbon emissions, for instance, electrification of heat and transport, leading to a higher utilisation of the network, whilst increasing dependency on an electricity supply. Consequently, the rising impact of power outages on economy and society has drawn attention to an emerging need for power network resilience. Resilience in terms of a power network is the ability to 'rapidly recover from disruptive events and adapt its operation and structure to prevent or mitigate the impact of similar events in the future' [5].

Preventive actions, such as topology switching, islanding, generator re-dispatching, and deployment of storage systems or mobile substations have been shown to efficiently prevent the uncontrolled propagation of cascading failures and to increase power network resilience to extreme events [2, 6–8]. Compared to physical network reinforcements, preventive actions are entirely operational, build on the intrinsic capabilities of a modern smart grid, make use of the availability of decentralised energy systems and distributed generation, and do not require investments in bulk infrastructure [9].

The decision whether to apply a preventive action, and, if several alternative actions are available, which one to apply, is a non-trivial classification problem, and must consider the stochastic nature of extreme events. Established resilience enhancement frameworks usually include an optimisation problem for a resilience metric, for example, the energy not supplied [8, 10, 11]. However, calculating these metrics and solving the optimisation problem is time-consuming, and often involves Monte Carlo (MC) simulations. This makes the previously proposed methodologies in many cases only feasible on a long-term planning horizon, but not for real-time decision-making. In [12], this obstacle is overcome using decision trees, a type of supervised machine learning (ML), recommending on preventive actions in real-time based on readily available predictors. In [13], a risk-based approach is used for real-time decision-making on when to island a power network, however, requiring extensive calculations of cascading failure impacts. Although these works show the feasibility of using ML for

decision-making on preventive actions, they are only capable of considering a specific preventive action and hazard. They thus lack the flexibility to consider a wider range of possible actions available to the network operator and different types of hazards that the system may be exposed to. Furthermore, they do not provide an evaluation of resilience benefits, for example, via established resilience metrics, and are thus challenging to compare to other resilience enhancement strategies.

The benefits of ML for real-time classification and decision-making have already been applied to a range of power system applications. Popular classification algorithms include logistic regression, naive Bayes, k-nearest neighbours, decision trees, and Support Vector Machines (SVMs). Applications in resilience improvements include, amongst others, predicting component outages during extreme weather events [14], cascading failure prediction [15], cascading failure prevention via congestion management [16], grid disaster mitigation via distributed energy resources [17], and self-healing [18]. However, research on decision-making for resilience enhancement of power systems using ML is limited so far, as three recent review papers suggest [19–21].

In this paper, a flexible decision-making process based on SVMs is presented to identify the ideal preventive action in real-time, preparing a power network in face of an extreme weather event. Although other ML approaches, such as deep learning, have gained increasing popularity, SVMs are still seen as state-of-the-art in many real-world classification applications [22, 23]. Compared to deep learning and other classification algorithms, SVMs are easy to implement, fast to train, and do not rely on large training data-sets for accurate results. Here, SVMs are trained using lost load data obtained from MC simulations of extreme weather events. Once trained, the proposed SVMs use only predictors that are readily available prior to an extreme event, such as event location, radius, intensity, network topology and load, thus require no further extensive simulations or optimisations. Different forms of preventive islanding are implemented as preventive actions, but the methodology is flexible and can also be extended to other preventive actions. The performance of the trained SVMs is compared to an ideal but time-consuming classifier, which always finds the optimum action, and a trivial but fast classifier, which always chooses the same action. Significant reductions in the risk of extreme events and improvements of power network resilience are assessed in a scenario-based approach via two metrics: accuracy in finding the ideal action for a particular event, and mismatch in load not supplied (LNS) compared to the ideal action. Each section is accompanied by a case study to illustrate the concepts.

The main contributions of this work are:

i. A probabilistic methodology for simulating and quantifying the risk of extreme events, such as windstorms, on power networks, based on AC-modelling of cascading failures.
ii. A resilience metric for evaluating and quantifying the effect of preventive actions (such as intentional islanding), balancing flexible pre-event demand response and post-event lost load.
iii. A risk-based decision-making framework for preventive actions, quantifying the performance of (ML) classifiers for identifying the optimum preventive actions for extreme events.
iv. An in-depth description and evaluation of a novel classifier based on SVMs for real-time decision-making on preventive actions improving power network resilience.

This paper is structured as follows: Section 2 describes the methodology for modelling extreme weather events and assessing their impact on power networks. Section 3 introduces the concept of preventive actions and presents the preventive actions implemented in this study. Section 4 proposes a decision-making process based on decision theory using a utility function for evaluating preventive actions. Due to the complexity of the methodology, each chapter is accompanied by a case study to illustrate the concepts. Section 5 introduces how SVMs can be applied to decision-making, describes the training and cross-validation of SVMs for decision-making on preventive actions, and presents a scenario-based performance assessment of SVMs. Figure 1 illustrates the individual steps of the methodology with reference to the respective sections of this paper.

2 | ASSESSING THE RISK OF EXTREME EVENTS

The following sections describe the individual steps of the methodology to assess the risk of extreme events (highlighted in blue in Figure 1), in detail. Although this study focuses on wind storms, the methodology can also be adapted to other hazards, for which forecasts are available and whose impact is describable via fragility curves. This includes, for example, flooding and heatwaves. Through advances in predicting cyber-attacks, the methodology may also be applied to mitigating the risk of such man-made threats [24, 25]. Furthermore, a combined risk assessment in case of multiple, coinciding hazards may be conducted. There are mainly two approaches that can be adopted in such combined risk analysis. The failure probabilities can be calculated for each asset and hazard individually, and then combined to a single failure probability per asset. Alternatively, if a combined fragility curve for assets is available that describes the failure probability to multiple, coinciding hazards, the methodology can be used without modifications by applying this combined fragility curve for the hazards to be considered in the analysis.

2.1 | Network model

Networks are modelled as sets of nodes and lines, representing busbars and the connections between them, respectively. Loads and generators can be present at every node, and are dispatchable, for example, due to implementation of demand-side management. This means that the actual amount of demand and supply at each node can be adjusted by the network operator, as long as they do not exceed their maximum load or generation

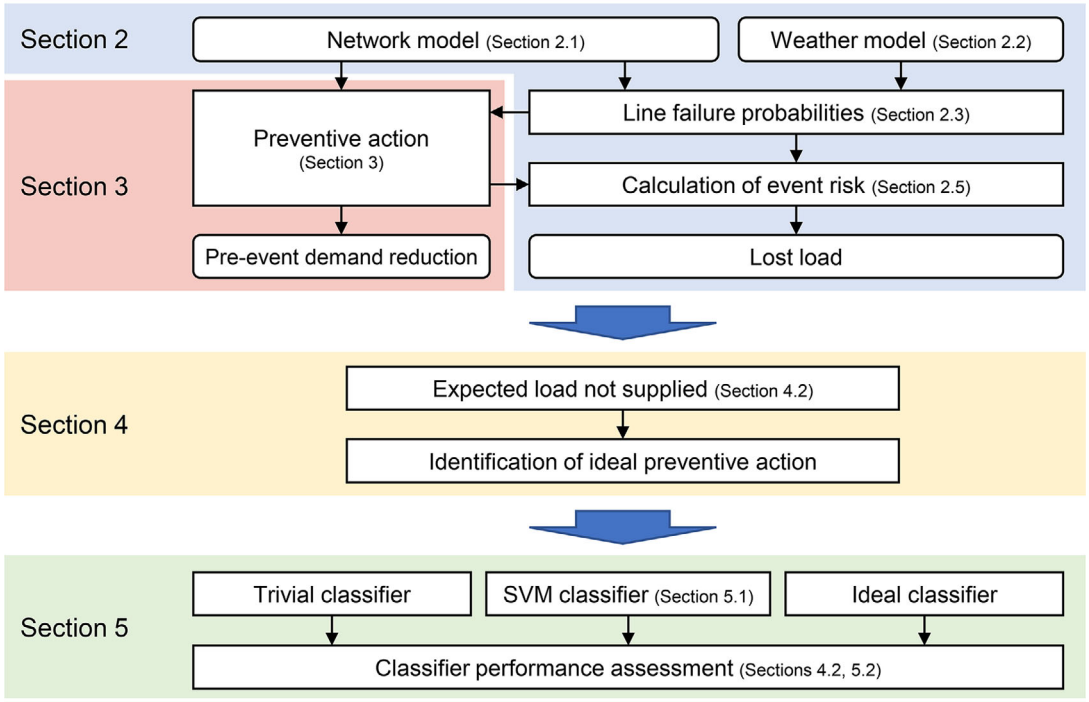
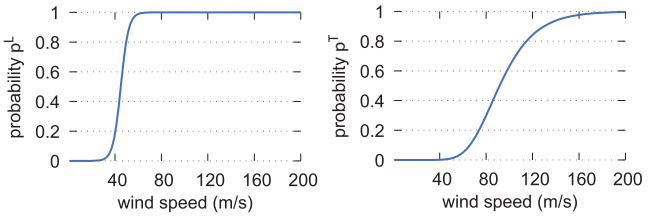


FIGURE 1 Methodology for training and evaluating an SVM for real-time selection of preventive actions

capacity, respectively. It is shown in Section 3 that the necessary amount of demand-side management is within the technically feasible demand response capacity of approximately 10% of the peak load [26, 27]. The geographic position of a node n is described by the vector $\boldsymbol{r}_n$. These positions are later used to identify which areas are exposed to an event. For each line (n, m), connecting nodes n and m, the maximum power rating and the length $L_{n,m}$ are given. The course of a line is described by a path function $\boldsymbol{\gamma}_{n,m} : [0, L_{n,m}] \to \mathbb{R}^2$, with $\boldsymbol{\gamma}_{n,m}(0) = \boldsymbol{r}_n$ and $\boldsymbol{\gamma}_{n,m}(L_{n,m}) = \boldsymbol{r}_m$. The path function can be a straight line or any other non-closed path.



(a) Line fragility curve per 100 km. (b) Tower fragility curve.

FIGURE 2 Wind fragility curves [5]. (a) Line fragility curve per 100 km. (b) Tower fragility curve

2.2 | Simulation of wind storms

A wind storm, such as a hurricane, typhoon, or a low-pressure system, can be characterised by a function $v_w(\boldsymbol{r})$ that returns the maximum observed wind speed, that is, gust speed, at location $\boldsymbol{r}$. v_w can in theory describe any type or extent of wind storm. For simplification, in this study v_w is given by

$$v_w(\boldsymbol{r}) = \begin{cases} \hat{v}_w, & \text{if } ||\boldsymbol{r} - \boldsymbol{r}_0|| \leq R \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

$\hat{v}_w$ is the maximum observed wind speed of the wind storm, or *intensity*, $\boldsymbol{r}_0$ is the geographic *location* of the centre of the wind storm, and R is the *radius* of the wind storm. These are parameters that can, for instance, be extracted from weather reports. More complex or non-circular wind systems can be described by a superposition of multiple wind storms or a different mathematical formulation, but do not alter the methodology of this study. Note that the maximum observed wind speed outside the wind storm can be set to zero even if this is not actually the case, as long as the maximum observed wind speed is below a wind speed at which damage to a power network can be expected. This expectancy is quantified in the next section using fragility curves.

2.3 | Line failure probabilities

Failure probabilities for overhead lines and towers are obtained from fragility curves (Figure 2), that give the fault probability of a tower or line segment depending on v_w. The total failure probability of a line (n, m) is

$$p_{n,m} = p^L_{n,m} + p^T_{n,m} - p^L_{n,m} \cdot p^T_{n,m}, \tag{2}$$

where $p^L_{n,m}$ and $p^T_{n,m}$ are the failure probabilities of the entire line and all towers in the line, respectively [5, 28]. It is now assumed that the fault of any tower or line segment in a line leads to a failure of the entire line. Instead of summing up the fault probabilities of all possible combinations of line segments or towers, the probability of the complementary case, that is when there is no damage at all and all line segments and towers survive, is calculated. This survival probability equals the product of all individual survival probabilities. With $n_{n,m} = \frac{L_{n,m}}{100\ \text{km}}$, the number of line segments in line (n, m), $p^L_{n,m}$ can thus be calculated by

$$p^L_{n,m} = 1 - \prod_{i=1}^{n_{n,m}} (1 - p^L(v_w(\gamma_{n,m}(i \cdot 100\ \text{km})))), \tag{3}$$

where p^L is the line fragility curve. Equation (3) can be transformed into a continuous expression via a geometric integral:

$$p^L_{n,m} = 1 - \exp \int_{\gamma_{n,m}} \log \left(1 - p^L(v_w(\gamma_{n,m}))^{\frac{1}{100\ \text{km}}}\right) \mathrm{d}s. \tag{4}$$

Likewise, with p^T the tower fragility curve, $T_{n,m}$ the set of all towers in line (n, m) and $\boldsymbol{r}^T_i$ the position of tower i, $p^T_{n,m}$ can be calculated by

$$p^T_{n,m} = 1 - \prod_{i \in T_{n,m}} \left(1 - p^T(v_w(\boldsymbol{r}^T_i))\right) \mathrm{d}s. \tag{5}$$

2.4 | Survival probability

For each event, a survival probability (SP) can be calculated that gives the probability of no line failure happening at all:

$$\text{SP} = \prod_{(n,m)} (1 - p_{n,m}). \tag{6}$$

SP combines event parameters and network topology in a single metric. SP can thus be used as a simple measure for the probability that an extreme events cause damage to a network, however, without considering the impact of said events.

2.5 | Calculation of wind storm risk

The risk of an event e is calculated using an MC simulation over a set S of event outcomes (cf. [5, 28, 29]). Each event outcome $s \in S$ comprises a set of lines damaged by the event. The event outcome is determined by comparing $p^{v_w}_{n,m}$ of each line to a random number $r^s_{n,m} \in [0, 1]$, which is sampled from a uniform distribution and generated for each event outcome and line individually, so that

$$s = \left\{ (n, m) \mid p^{v_w(e)}_{n,m} > r^s_{n,m} \right\}. \tag{7}$$

Damaged lines that are no longer operational can lead to overloading of parallel routes, which are tripped by the protection system. At worst, this triggers a cascading failure and leads to dynamic instability and disintegration of the network. Dedicated cascading failure models are widely presented in the literature, and the proposed methodology can implement a range of available models [30]. The choice of cascading failure model has significant influence on the subsequent decision-making, and it is important to use a model that considers the mechanisms that play a crucial role in cascade propagation. Cascading failures are simulated using a quasi-steady-state AC cascading failure model[1] that considers overload protection of transmission lines as well as static representations of frequency and voltage protection mechanisms and excitation limiters [31]. This model has previously been shown to produce accurate and validated results following recommendations by the IEEE PES working group on cascading failures in a computationally efficient and robust way [31, 32]. The MC simulation returns a probability distribution of lost load P^{LL}_e with mean $\hat{P}^{\text{LL}}_e$ and standard deviation σ^{LL}_e in the network after event e.

P^{LL}_e depends largely on the network topology, load, and generator dispatch. P^{LL}_e can be influenced by applying preventive actions (highlighted in red in Figure 1), which is described in Section 3.

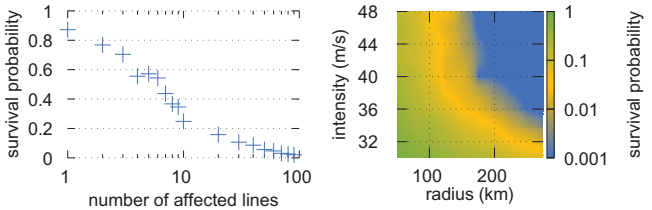
2.6 | Case study

The methodology for assessing the risk of extreme events (highlighted in blue in Figure 1 and described in the previous sections) was tested on an AC model of the German transmission network, as specified by the SciGRID project [33]. The network, reduced to high and extra high voltage level, consists of 489 buses and 825 lines. Locations and characteristics of loads and generators were taken from snapshots provided by the PyPSA toolbox [34] and scaled up to a peak load of 80 GW, approximately matching German peak demand [35]. The load is later set to random values between 50% and 100% of peak load to simulate various network conditions.

Note that 20,000 random events were created by sampling the event parameters location, radius, and intensity from a quasi-random Halton set. Compared to uniformly distributed random numbers, quasi-random numbers offer the benefit of covering the space of event parameters more evenly and with lower discrepancy [36]. $R \in [50, 300]$ km and $\hat{v}_w \in [30, 50]$ m/s, although other ranges could have been chosen without affecting the methodology.

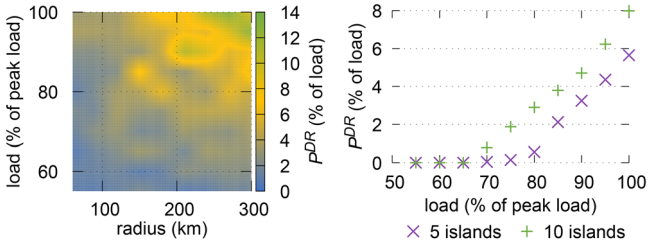
SP was calculated for each event individually. SP was high if an event hit in an area without any lines, and decreased gradually with increasing number of affected lines (Figure 3a). Likewise, SP was high if either event radius or intensity were small, and

[1] Source code for AC cascading failure model is publicly available from https://github.com/mnoebels/AC-CFM.



(a) Affected lines. (b) Radius and intensity.

FIGURE 3 Survival probability. (a) Affected lines. (b) Radius and intensity



(a) Single area. (b) Islanding.

FIGURE 4 Pre-event demand reduction with preventive actions. (a) Single area. (b) Islanding

decreased gradually with increasing radius or intensity (Figure 3b). This matches previous expectation and demonstrates that SP can be used as a simple measure for the probability that an extreme events cause damage to a network.

3 | PREVENTIVE ACTIONS

The following sections introduce the concept of preventive actions for improving power network resilience, and propose a resilience-based metric for evaluating the effect of preventive actions, exemplified via intentional islanding. Figure 1 illustrates (highlighted in red) how preventive actions sit within the methodology of this paper. Preventive actions are operational measures that are applied by the network operator before an event causes severe degradation. They can consider weather forecasts, demand predictions, and network topology, and can thus be tailored to the specific conditions at the time of an event. Preventive actions include, for instance, temporarily re-dispatching loads and generators, reconfiguring the network, or sectionalizing the network into islands. Such actions are achieved by using existing smart-grid capabilities and require no additional investments in bulk electrical power infrastructure. Preventive actions can be assumed to not affect the dynamic stability of the network as there is sufficient time for the network operator to re-dispatch the network and manage the implementation of the actions in a controlled manner. This is in contrast to corrective actions which act suddenly and may leave the network in a volatile, and potentially unstable, state. In this study, the network is hence re-dispatched after applying preventive actions using an optimum power flow solver. However, a preventive action may temporarily require a pre-event demand reduction P^{DR}, if demand exceeds generation capacity. In practice, P^{DR} can be achieved through controllable demand management (e.g. switching off big industrial or commercial loads). Note that P^{DR} can be accurately calculated based on the current generation capacity and demand in the network, the network topology, availability of transmission lines, and phasor measurement data.

Once the network is hit by an event, a preventive action may prevent the uncontrolled propagation of cascading failures, and thus reduces the lost load. As discussed in Section 2.5, P_e^{LL} is a probability distribution, because the actual outcome of an event is a stochastic process. Thus, the LNS P_e^{LNS} due to an event, summarising P^{DR} and P_e^{LL}, is also a probability distribution and given by

$$P_e^{\mathrm{LNS}} = P_e^{\mathrm{DR}} + P_e^{\mathrm{LL}}. \tag{8}$$

3.1 | Preventive islanding

The preventive actions considered are isolating a single area (area) and islanding. These actions contain cascading failures within a part of the network. Islanding is performed via spectral clustering [37], using power flows to construct the Laplacian matrix. This groups buses in a way that minimises load flow interruptions whilst creating islands of similar size. The number of islands to be created needs is defined by the network operator. Islands depend on network topology and load flows, but do not consider event parameters. Isolating a single area is an alternative preventive actions, considering event parameters and aiming to separate the parts of a network that are affected by an event in a single island, whilst retaining the rest of the network. This is performed via constrained spectral clustering [38]. Both spectral clustering and constrained spectral clustering are established methods in the literature for intentional, preventive islanding [5, 39].

3.2 | Case study

The characteristics of preventive actions are investigated in a case study, simulating 1000 randomly created events in the previously introduced German transmission network.

P^{DR} for isolating a single area depended mainly on load and event radius (Figure 4a). With increasing load and radius, P^{DR} increased, but stayed mainly below 10% except for a network at peak demand and events with radius 300 km.

P^{DR} for islanding depended mainly on load and the number of islands (Figure 4b). Until a certain load threshold, P^{DR} was close to zero, but increased gradually with load for loads higher than the threshold. The threshold became smaller with more islands. P^{DR} for islanding was independent of the event radius, because the islands were created purely based on load and irrespective of the event. P^{DR} stayed well below 10%.

In the following, for each event, 100 event outcomes were modelled. This was assumed to be sufficient for converging

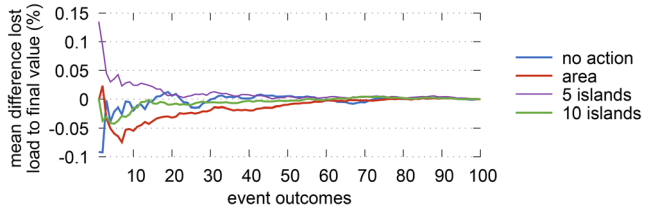
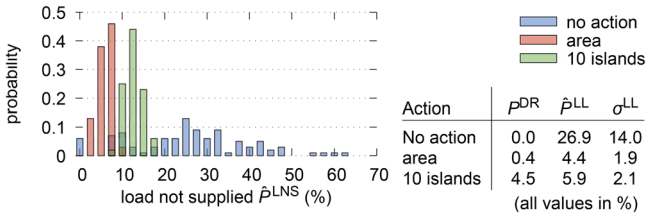


FIGURE 5 Convergence of lost load $\hat{P}^{LL}$ for each action considered



Action	P^{DR}	$\hat{P}^{LL}$	σ^{LL}
No action	0.0	26.9	14.0
area	0.4	4.4	1.9
10 islands	4.5	5.9	2.1

(all values in %)

FIGURE 6 LNS for an exemplary event with different preventive actions

mean lost load for all actions considered as a convergence analysis previously showed (Figure 5).

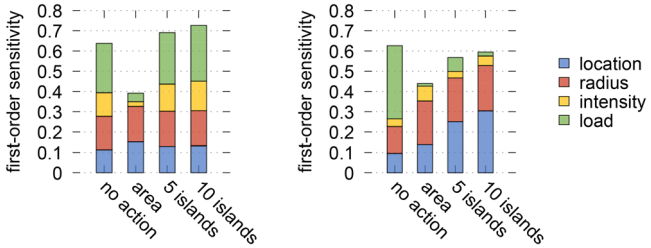
For the purpose of demonstration, Figure 6 compares P^{LNS} of a single, exemplary event without preventive action, with isolating a single area, and with islanding, for 100 event outcomes. The probability distribution of P^{LNS} with preventive actions applied does not start at 0%, but has a small onset due to P^{DR}. However, because the propagation of cascading failures was restricted, P^{LNS} with preventive actions applied has, in this case, a narrower probability distribution and is centred at lower values than without preventive action. This can also be observed from the values for $\hat{P}^{LL}$ and σ^{LL}.

Different events affect a network in different ways, and network topology and load at the time of the event further influence whether a preventive action has a positive impact on P_e^{LNS}. It can be concluded from the case study that many parameters need to be considered in any decision-making on preventive actions.

4 | DECISION-MAKING ON PREVENTIVE ACTIONS

The following sections describe a risk-based decision framework for preventive actions and introduce a performance metric for evaluating classifiers deciding on preventive actions (highlighted in yellow in Figure 1). The decision when to apply a preventive action, and, if several alternative preventive actions are available, which one to apply, is non-trivial. Although P_e^{DR} can usually be accurately calculated, P_e^{LL} is a probability distribution and as such the actual lost load after an event is based on a stochastic process.

Decision theory aims at identifying the optimal decision, particularly in the presence of such uncertainty. Following the considerations of Bernoulli and the Von Neumann-Morgenstern utility theorem [40], the choice between different actions leads to different 'lotteries'. Each lottery can be assigned to an



(a) Mean lost load $\hat{P}^{LL}$. (b) Standard deviation σ^{LL}.

FIGURE 7 First-order sensitivity coefficients of lost load and its standard deviation calculated by FAST (missing values to one are higher order interactions). (a) Mean lost load $\hat{P}^{LL}$. (b) Standard deviation σ^{LL}

expected utility, a measurable preference of an outcome. Several metrics can be included in the expected utility, such as $\hat{P}_e^{LL}$, P_e^{DR}, the Value of Lost Load, the number and type of customers affected, the outage duration, and others. Under some circumstances, it might also be important to consider the σ_e^{LL}, for instance, if preference should be given to actions that minimise the variety of possible outcomes. The ΦΛΕΠ resilience metric framework proposes a number of further resilience-based metrics [41]. Within this framework, P_e^{LL} is equivalent to the Λ metric for operational resilience.

4.1 | Sensitivity analysis

To determine which event parameters influence P_e^{LL}, a sensitivity analysis for $\hat{P}_e^{LL}$ and σ_e^{LL} was performed using a Fourier Amplitudes Sensitivity Test (FAST) [42]. FAST calculates the first-order sensitivity coefficients that describe to what percentage the variance of mean lost load and standard deviation is related to the event parameters. FAST is a global sensitivity test, meaning that the sensitivity of an event parameter is averaged over variations in the other event parameters. Higher orders, that is, interactions between event parameters, are ignored. FAST was performed using GSAT for MATLAB [43].

In the previously introduced German transmission network, mean lost load and standard deviation were, for all actions, dominated by first-order sensitivities (Figure 7). Lost load when isolating a single area was significantly less sensitive to event intensity and load, but slightly more sensitive to event location compared to other actions. This demonstrates the strength of this action to contain cascading failures within the isolated area, no matter how badly this area was affected by an event, as long as the location of the event is known. Less sensitivity on event location compared to isolating a single area could be achieved, for instance, by islanding, but with a higher sensitivity to event intensity and load. Preventive actions greatly reduced the sensitivity of the standard deviation of lost load to load, but increased the sensitivity to event location. Most importantly, the sensitivity analysis underlines that multiple parameters significantly influence lost load and its standard deviation, confirming that any process trying to optimise lost load is non-trivial and needs to consider all these event parameters.

4.2 | Expected load not supplied, trivial and ideal decision

As discussed in Section 4, decision theory requires an expected utility for consideration in a decision-making process. Here, the expected utility of event e and action a is the *expected load not supplied* (ELNS):

$$\hat{P}_e^a = P_e^{a,\mathrm{DR}} + \hat{P}_e^{a,\mathrm{LL}}. \tag{9}$$

Note that (9) can also take other forms, for example, if $P_e^{a,\mathrm{DR}}$ and $\hat{P}_e^{a,\mathrm{LL}}$ should be weighted differently. Such considerations, however, do not change the following methodology. $\hat{P}_e^{a,\mathrm{LL}}$ is a direct outcome of the simulation of cascading failures (cf. Section 2.5). As such, (9) considers the state of the power network after handling all cascade propagation mechanisms that are implemented in the cascading failure model used. The cascading failure model used in this paper considers, for example, overload protection, over- and under-excitation limiters, and voltage and frequency protection.

A decision-making process is defined by a classifier $C : E \to A$ that assigns an event $e \in E$ to an action $a \in A$, where E is the set of events and A is the set of actions. The performance of a classifier C can be assessed by the *mean ELNS*, averaged over all events in E:

$$\hat{P}^C = \frac{1}{N_e} \sum_{e \in E} \hat{P}_e^{C(e)}. \tag{10}$$

Here, N_e is the number of events in E. A low mean ELNS corresponds to a high classifier performance.

Some classifiers can easily be defined. A *trivial* classifier $C_{\mathrm{trivial}}(e)$ assigns the same action to every event. Although this is computationally very efficient, the trivial classifier might certainly does not perform well for any possible events. Note that every available action leads to a trivial classifier returning the respective action for any event.

The *ideal* classifier, which minimises $\hat{P}^C$, assigns to every event the action that leads to the minimum ELNS:

$$C_{\mathrm{ideal}}(e) = \arg\min_a \hat{P}_e^a. \tag{11}$$

This leads to the ideal mean ELNS:

$$\hat{P}^{\mathrm{ideal}} = \frac{1}{N_e} \sum_e \hat{P}_e^{C_{\mathrm{ideal}}(e)} = \frac{1}{N_e} \sum_e \min_a \hat{P}_e^a. \tag{12}$$

Obtaining the ideal classifier requires knowledge of $\hat{P}_e^a$ for all $a \in A$. The ideal classifier thus performs best, but is also very time-consuming.

4.3 | Case study

To demonstrate the concept of mean ELNS and the complexity of decision-making, $\hat{P}^a$ was calculated for all actions using an MC-simulation of $N_e = 1000$ events. The parameters, that is, event location, radius, intensity and load, of each event were sampled from a Halton set. For all trivial classifiers, that is, all individual actions, ELNS increased with increasing load (Figure 8, events grouped into bins by load) or decreasing SP (Figure 9, events grouped into bins by SP), independent of the action. Until a load of 93% of peak load, 'no action' led to the lowest mean ELNS. For higher loads, ELNS for 'no action' increased sharply, while for preventive actions it increased less steep, with 'area' leading to the lowest mean ELNS. This was expected considering that higher load leads to higher stress on the network and an increased risk of cascading failures. There was no obvious preference of actions depending on SP.

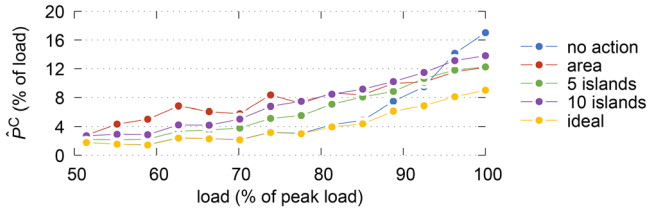


FIGURE 8 Mean ELNS depending on load for trivial and ideal classifiers

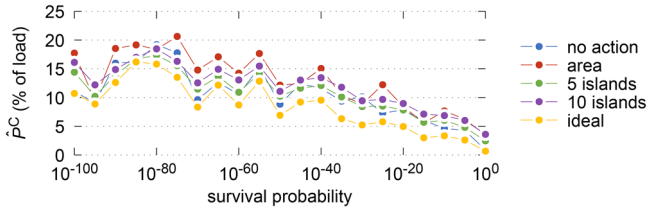


FIGURE 9 Mean ELNS depending on SP for trivial and ideal classifiers

For a load of up to 80% of peak load, the mean ELNS of the ideal classifier (yellow in Figure 8) closely follows the curve for 'no action', suggesting that for a load of less than 80% the ideal action was in most cases 'no action'. For a load of larger than 80% of peak load, the ideal curve does not follow any of the curves of the trivial classifiers. When looking at SP (Figure 9), the ideal classifier does not follow any of the trivial classifiers but is significantly lower than any of them. This shows that the ideal classifier, that is, finding the ideal action for each event, significantly reduces the ELNS compared to any trivial classifier, that is individual action. Particularly, the ideal action cannot only be determined by load or SP, but also depends on the other event parameters. This has already been suggested by the sensitivity analysis (Section 4.1).

In the previous considerations, $C_{\mathrm{ideal}}(e)$ was determined based on a time-consuming MC simulation of a large number of events and event outcomes. In practice, there is usually not enough time for such extensive simulations to be run in face of an upcoming event. The challenge thus is to find a time-efficient classifier with good performance.

5 | SUPPORT VECTOR MACHINES

Section 4 showed that identifying the ideal action for an event is a non-trivial decision-making process, which needs to include multiple parameters and usually relies on time-consuming simulations, making it less attractive and feasible for real-time application. In this section (highlighted in green in Figure 1), it is shown that an SVM can be trained to reliably predict the ideal action based on the event parameters location, radius and intensity, the network topology and load. These event parameters are readily available prior to an event, for instance, from weather reports, phasor data, or smart meters. The performance of the SVM, evaluated using the performance metric introduced in Section 4, is compared to an ideal and a trivial classifier (cf. Section 4.2).

An SVM is a supervised learning approach for classification of data-points [44]. An SVM is trained using a set of training data-points, containing an arbitrary number of predictors, with known classes. The data points in this study are the event parameters, or predictors, and the classes are the ideal actions for each event. Kernel functions can be used for data points which are not linearly separable by transferring data-points into a higher dimensional feature space, however, potentially increasing the generalisation error and the risk of overfitting. Multi-class fitting is possible via error-correcting output codes (ECOCs) [45]. The *fitcecoc* routine of MATLAB 2019a is used for SVM training.

5.1 | Feature selection

SVMs require the predictors, or features, to be selected before training. This is in contrast to other ML techniques, such as deep learning, where features can be selected automatically. The aim is to select features that are readily available prior to an event and which will result in high SVM performance in later predictions. Particularly, appropriate SVM feature selection results in high SVM performance, for example, a high accuracy, although a high SVM performance does not necessarily mean that all features have the same importance.

The sensitivity analysis in Section 4.1 already established a range of parameters that largely govern the behaviour of lost load after an event, and which are hence selected as features. These are: event location, event radius, event intensity, and network load. Additionally, survival probability (cf. Section 2.4) is added as an additional important feature which aggregates information about the topology of the network. When following the proposed methodology for modelling extreme events, these features can be calculated independently of the network and type of hazard, highlighting the flexibility of the approach taken in this paper.

Note that SVMs are assessed in this paper for individual hazards only. In the case of multiple, coinciding hazards, each hazard needs to be expressed by its own individual event parameters. These may also vary for different types of hazards (e.g. wind speed or flooding level). The training should then be performed using a larger set of features that include these additional event parameters.

5.2 | Training and cross-validation

In this section, the training and cross-validation of SVMs is described. The training set consists of $N_e = 1000$ events. Each event is characterised by the event parameters location, radius, intensity, and load. These parameters are sampled from a Halton sequence to cover the space of event parameters more evenly. As discussed in the previous section, the event parameters and SP, which can be calculated based on the event parameters, are used as predictors. For each event, the ideal action is determined based on (11), that is, the action that leads to the minimum ELNS. ELNS is calculated for all events and actions using an MC simulation of 100 event outcomes, that is, randomly generated sets of lines damaged by the event. This number has previously been shown to be sufficient for convergence (cf. Section 3, Figure 5). The training data is made publicly available [46].

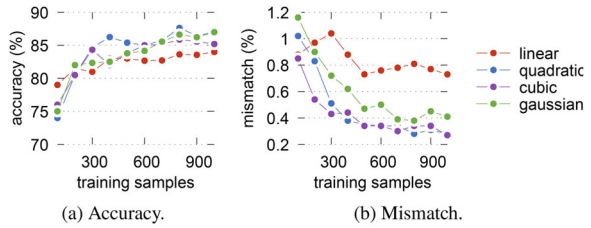
The performance of an SVM $C_{\mathrm{SVM}}(e)$, which returns the predicted action for event e, is evaluated using two metrics: Accuracy is the percentage of events in which the SVM correctly predicted the ideal action. Note that the accuracy of the ideal classifier is 100%. However, here it is not necessarily required to achieve a high accuracy, as long as the ELNS when choosing the action predicted by the SVM is close to the ELNS of the ideal classifier. Thus, ELNS mismatch is used as an alternative metric, which is the mean ELNS difference between predicted and ideal action for all events:

$$\text{mismatch} = \hat{P}^{\mathrm{SVM}} - \hat{P}^{\mathrm{ideal}}, \text{ with} \tag{13}$$

$$\hat{P}^{\mathrm{SVM}} = \frac{1}{N_e} \sum_{e} \hat{P}_e^{D(e)}. \tag{14}$$

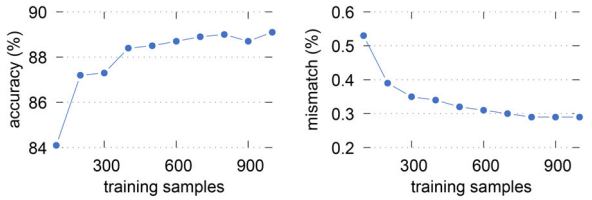
The larger the mismatch, the less the performance of the SVM. Although accuracy is an established performance metric in ML applications, ELNS mismatch directly relates to the earlier defined expected utility has a clear meaning to the network operator. ELNS mismatch should hence be seen as the primary performance metric here.

A 10-fold cross-validation with a hold-out of 10% is used to identify the optimal kernel function. Furthermore, through cross-validation overfitting and generalisation errors can be avoided [47]. This means that for each kernel function, 10 SVMs are trained using a random 90% of events from the training set, and accuracy and mismatch are calculated for the remaining 10% of events. The arithmetic mean of accuracy and mismatch is then calculated for each kernel function separately. The kernel functions used in this study are: linear, quadratic, cubic, and Gaussian. Furthermore, the dependence of SVM performance on number of training samples is investigated by using only a subset of the entire training set. This is important as the number of training samples directly relates to the time needed for generating the training set. The impact of number of training samples on SVM training time is negligible (Section 5.3.2).



(a) Accuracy. (b) Mismatch.

FIGURE 10 SVM cross-validation depending on number of training samples. (a) Accuracy. (b) Mismatch



(a) Accuracy. (b) Mismatch.

FIGURE 11 SVM performance depending on number of training samples. (a) Accuracy. (b) Mismatch

For all kernel functions, accuracy increases with increasing number of training samples (Figure 10a). Accuracy of more than 80% is achieved for 200 or more training samples independent of kernel function. For quadratic, cubic, and Gaussian kernel functions, the increase in accuracy is most significant for up to 300 training samples. Accuracy still increases for higher numbers of training samples, but not as significant. The highest accuracy for 300 or more training samples is achieved using a quadratic kernel function.

Similarly, mismatch decreases with increasing number of training samples for all kernel functions (Figure 10b). Again, decrease in mismatch for quadratic, cubic, and Gaussian kernel functions is most significant for up to 300 training samples, and less significant for more than 300 training samples. Accuracy and mismatch is the lowest using a quadratic kernel function for 400 or more training samples. Based on the cross-validation, a quadratic kernel function is selected in the following.

5.3 | SVM performance assessment

After cross-validation and identification of the best kernel function (quadratic), an SVM is first trained on the training set, and then tested on a test set consisting of $N_e = 1000$ events, which are different from the events in the training set. The following presents a performance analysis of the proposed SVM-based method on four aspects: Firstly, the relationship between SVM performance, measured via accuracy and mismatch, and the number of required training samples is analysed. Secondly, the computational efficiency of the SVM is studied by comparing the required time for training the SVM to the time taken for classification with the ideal classifier. Thirdly, the dependency of SVM performance on network load and SP is investigated. Finally, a scenario-based analysis is undertaken to study SVM performance in more detail under different critical conditions, such as high load or high impact.

5.3.1 | Training samples

First, the impact of the number of training samples on SVM performance in the test set is analysed. Accuracy for the test set increases with increasing number of training samples (Figure 11a). The increase in accuracy is most significant for up to 400 training samples, and settles for higher numbers of training samples at approximately 88.8%. Mismatch of the training set decreases with increasing number of training samples (Figure 11b). Again, mismatch first decreases significantly for up to 300 training samples, and less significantly for higher numbers of training samples. This matches previous observations when cross-validating SVMs (Section 5.2), and demonstrates that the gain of increasing the training set size is limited. Particularly, depending on desired accuracy and mismatch, training set size may be reduced in order to speed up generation of the training set. In the following, the entire training set (1000 samples) is used.

5.3.2 | Computation and classification times

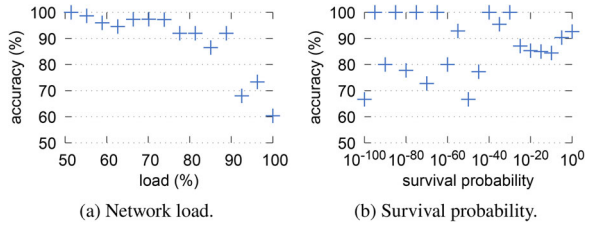
Classification times for the ideal classifier and the SVM were determined using a machine with a 8 core Intel Xeon E5-2620 CPU and 32 GB RAM. The ideal classifier had a mean classification time of 7.98 minutes per event. This is the same as the computation time required to generate one training event for the SVM. After the complete training set containing 1000 events was obtained, SVM training took 0.17 seconds. Once the SVM was trained, classification of an unknown event took 0.005 s on average. This relates to a classification time that is more than three orders of magnitude shorter compared to the ideal classifier. Note that for the SVM, generating the training set and SVM training can be done in advance. The significant shorter classification time compared to the ideal classifier shows that the proposed SVM approach is in fact capable for real-time decision-making.

5.3.3 | Accuracy and mismatch

Assessing the performance of the SVM depending on network load reveals that SVM accuracy decreases for higher network load (Figure 12a, averaged over all other event parameters). There is no obvious dependency of accuracy on SP (Figure 12b). A lower accuracy is not necessarily an issue, as long as the ELNS mismatch is small (cf. Section 5.2). The ELNS mismatch can be observed when looking at the ELNS for each trivial classifier, that is, individual action, as well as the ELNS of the ideal classifier and the SVM. For both load and SP, the ELNS for the SVM follows the ideal ELNS closely even for higher loads and lower survival probabilities, indicating a small mismatch (Figures 13 and 14). Particularly, ELNS for the SVM is always lower than

TABLE 1 Scenario-based performance evaluation of SVM

			ELNS (%)						σ^{ELNS} (%)		
Scenario	**Events considered**	N_e	**no action**	**area**	**5 islands**	**10 islands**	**ideal**	**SVM**	**no action**	**SVM**	**Accuracy**
All		**1 000**	**5.3**	**8.1**	**6.3**	**7.4**	**4.2**	**4.4**	**8.7**	**7.2**	**89.1%**
High load	Load > 80%	400	9.6	10.9	10.0	11.2	6.7	7.7	11.1	8.9	78.3%
Low SP	SP < 5%	773	6.8	9.8	7.6	8.8	5.3	5.6	9.4	7.8	87.5%
No impact	ELNS < 1%	532	0.2	3.2	1.8	2.9	0.2	0.2	0.3	0.3	96.2%
Low impact	1% ≤ ELNS < 5%	171	2.7	7.4	5.4	6.7	2.5	2.6	1.2	1.2	91.2%
Medium impact	5% ≤ ELNS < 20%	212	10.8	13.8	11.9	13.5	9.1	9.7	4.2	4.8	79.3%
High impact	ELNS ≥ 20%	85	29.1	25.9	22.3	22.5	20.0	21.6	5.4	8.6	64.7%



(a) Network load. (b) Survival probability.

FIGURE 12 SVM accuracy. (a) Network load. (b) Survival probability

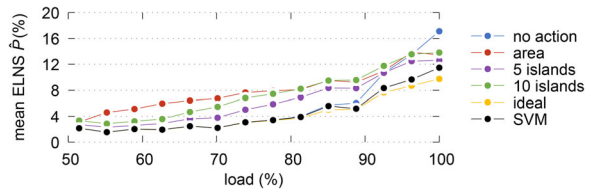


FIGURE 13 Comparison of ELNS depending on load for different individual actions, ideal actions, and actions predicted by SVM in the test set

any of the individual actions. This means that the SVM reliably performs better than any trivial classifier, that is, always choosing the same action.

5.3.4 | Scenario-based performance analysis

To further investigate and quantify the performance of the SVM under various circumstances, events are grouped into scenarios based on their characteristics. The scenarios considered are

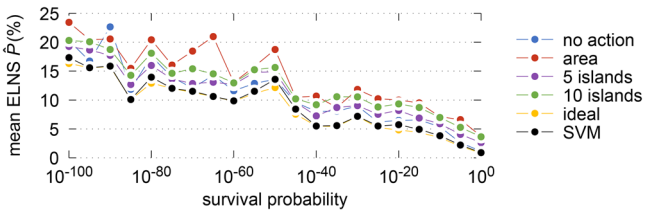


FIGURE 14 Comparison of ELNS depending on survival probability for different individual actions, ideal actions, and actions predicted by SVM in the test set

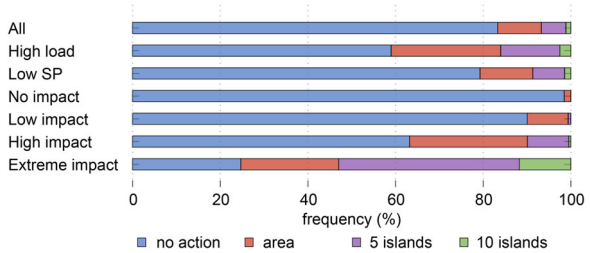


FIGURE 15 Frequency of actions being predicted by the SVM for every scenario

high load, low SP, no impact, low impact, medium impact, and high impact (Table 1). This is compared to considering all events (bold in Table 1). Note that events may fall into several scenarios. ELNS is calculated for each scenario and action, as well as the ELNS of the ideal classifier and when choosing the action predicted by the SVM. ELNS for the SVM is in every scenario lower than the ELNS for any individual action. This demonstrates, even though the SVM does not perform ideal, the increased performance of the SVM compared to any trivial classifier. Standard deviation of ELNS is also reduced for SVM compared to no action (Table 1) except for the medium impact and high impact scenarios. Note that this is purely an effect of the reduced standard deviation σ_e^{LL} for preventive actions, as discussed in Section 3, and σ_e^{LL} is not part of the definition of the ideal action in (11). Thus, further reductions in the standard deviation could be achieved by extending this definition, however, this was not part of this study.

Figure 15 shows the frequency each action is predicted by the SVM in each scenario. Particularly for events with a high load, high impact, or extreme impact, preventive actions are more often selected than on average. This shows the strength of preventive actions to improve power network resilience under extreme conditions or exposed to serious threats.

6 | CONCLUSION

Preventive actions to increase power network resilience to extreme events are extensively researched, but the decision if and how to apply a preventive action in face of an upcoming

event is non-trivial. In this paper, an SVM was successfully trained via MC simulations to work as a decision-making process for preventive actions such as islanding. Once the SVM was trained, the decision was only based on predictors that were readily available prior to an extreme event, including event location, radius and intensity, SP, and load. Using two metrics, accuracy and mismatch, the performance of the proposed SVM was assessed in a scenario-based approach, showing that it performed close to an ideal classifier even under extreme circumstances whilst featuring significantly short classification times. This makes the presented methodology a promising real-time decision-making tool for network operators to increase power network resilience.

ACKNOWLEDGEMENTS

The authors acknowledge financial support from EPSRC (EP/L016141/1 and EP/R030294/1) through the Power Networks Centre for Doctoral Training and the TERSE ('Techno-Economic framework for Resilient and Sustainable Electrification') project, and the Newton Prize project 'Resilient Planning of Low-Carbon Power Systems' under Project 1304.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Figshare at http://doi.org/10.48420/14784570, reference number [46].

ORCID

Matthias Noebels https://orcid.org/0000-0003-2330-3015
Robin Preece https://orcid.org/0000-0003-1224-0397
Mathaios Panteli https://orcid.org/0000-0003-4274-529X

REFERENCES

1. Lin, Y., Bie, Z., Qiu, A.: A review of key strategies in realizing power system resilience. Glob. Energy Interconnect. 1(1), 70–78 (2018)
2. Jufri, F.H., Widiputra, V., Jung, J.: State-of-the-art review on power grid resilience to extreme weather events: definitions, frameworks, quantitative assessment methodologies, and enhancement strategies. Appl. Energy 239, 1049–1065 (2019)
3. Panteli, M., Mancarella, P.: Influence of extreme weather and climate change on the resilience of power systems: impacts and possible mitigation strategies. Electric Power Syst. Res. 127, 259–270 (2015)
4. Ratnam, E.L., Baldwin, K.G.H., Mancarella, P., Howden, M., Seebeck, L.: Electricity system resilience in a world of increased climate change and cybersecurity risk. Electricity J. 33(9), 106833 (2020)
5. Panteli, M., Trakas, D.N., Mancarella, P., Hatziargyriou, N.D.: Boosting the power grid resilience to extreme weather events using defensive islanding. IEEE Trans. Smart Grid 7(6), 2913–2922 (2016)
6. Mureddu, M., Caldarelli, G., Damiano, A., Scala, A., Meyer-Ortmanns, H.: Islanding the power grid on the transmission level: less connections for more security. Sci. Rep. 6, 1–11 (2016)
7. Noebels, M., Panteli, M.: Assessing the effect of preventive islanding on power grid resilience. In: 2019 IEEE Milan PowerTech, pp. 1–6. Piscataway, NJ: IEEE (2019)
8. Taheri, B., Safdarian, A., Moeini-Aghtaie, M., Lehtonen, M.: Distribution systems resilience enhancement via pre- and post-event actions. IET Smart Grid 2(4), 549–556 (2019)
9. Bie, Z., Lin, Y., Li, G., Li, F.: Battling the extreme: a study on the power system resilience. Proc. IEEE 105(7), 1253–1266 (2017)
10. Ma, S., Chen, B., Wang, Z.: Resilience enhancement strategy for distribution systems under extreme weather events. IEEE Trans. Smart Grid 3053(c), 1–1 (2016)
11. Salomon, J., Broggi, M., Kruse, S., Weber, S., Beer, M.: Resilience decision-making for complex systems. ASCE-ASME J. Risk Uncertainty Eng. Syst. 6(2), 1–11 (2020)
12. Senroy, N., Heydt, G.T., Vittal, V.: Decision tree assisted controlled islanding. IEEE Trans. Power Syst. 21(4), 1790–1797 (2006)
13. Fernández-Porras, P., Panteli, M., Quirós-Tortós, J.: Intentional controlled islanding: when to island for power system blackout prevention. IET Gener. Transm. Distrib. 12(14), 3542–3549 (2018)
14. Eskandarpour, R., Khodaei, A.: Machine learning based power grid outage prediction in response to extreme events. IEEE Trans. Power Syst. 32(4), 3315–3316 (2017)
15. Gupta, S., Kambli, R., Wagh, S., Kazi, F.: Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework. IEEE Trans. Ind. Electron. 62(4), 2478–2486 (2015)
16. Zarrabian, S., Belkacemi, R., Babalola, A.A.: Reinforcement learning approach for congestion management and cascading failure prevention with experimental application. Electric Power Syst. Res. 141, 179–190 (2016)
17. Maharjan, L., Ditsworth, M., Niraula, M., Narvaez, C.C., Fahimi, B.: Machine learning based energy management system for grid disaster mitigation. IET Smart Grid 2(2), 172–182 (2019)
18. Karim, M.A., Currie, J., Lie, T.T.: Distributed machine learning on dynamic power system data features to improve resiliency for the purpose of self-healing. Energies 13(13), 1–20 (2020)
19. Ibrahim, M.S., Dong, W., Yang, Q.: Machine learning driven smart electric power systems: Current trends and new perspectives. Appl. Energy 272, 115237 (2020)
20. Alimi, O.A., Ouahada, K., Abu-Mahfouz, A.M.: A review of machine learning approaches to power system security and stability. IEEE Access 8, 113512–113531 (2020)
21. Xie, J., Alvarez-Fernandez, I., Sun, W.: A review of machine learning applications in power system resilience. 2020 IEEE Power & Energy Society General Meeting, pp. 1–5. Piscataway, NJ: IEEE (2020)
22. Bertsimas, D., Dunn, J., Pawlowski, C.: Robust classification. INFORMS J. Optim. 1(1), 2–34 (2019)
23. Sarker, I.H.: Machine learning: algorithms, real-world applications and research directions. SN Comput. Sci. 2(3), 1–21 (2021)
24. Zhan, Z., Xu, M., Xu, S.: Predicting cyber attack rates with extreme values. IEEE Trans. Inf. Forensics Secur. 10(8), 1666–1677 (2015)
25. Fang, X., Xu, M., Xu, S., Zhao, P.: A deep learning framework for predicting cyber attacks rates. Eurasip J. Inf. Secur. 5, 1–11 (2019)
26. Jahn, A., Gottstein, M.: Nachfragesteuerung im deutschen Stromsystem – die unerschlossene Ressource für die Versorgungssicherheit. RAP (2013)
27. US Federal Energy Regulatory Commission. Assessment of Demand Response & Advanced Metering. Washingtob, DC: FERC (2015)
28. Poudel, S., Dubey, A., Bose, A.: Risk-Based probabilistic quantification of power distribution system operational resilience. IEEE Syst. J. 14(3), 3506–3517 (2020)
29. Noebels, M. & Panteli, M. : Time series analysis of preventive islanding as a measure to boost power grid resilience. 2019 IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT) 1–5 (2019)
30. Guo, H., Zheng, C., Iu, H.H.C., Fernando, T.: A critical review of cascading failure analysis and modeling of power system. Renewable Sustain. Energy Rev. 80, 9–22 (2017)
31. Noebels, M., Preece, R., Panteli, M.: AC cascading failure model for resilience analysis in power networks. IEEE Syst. J. 1–12 (2020)
32. Bialek, J., Ciapessoni, E., Cirio, D., Cotilla-Sanchez, E., Dent, C., Dobson, I., et al.: Benchmarking and validation of cascading failure analysis tools. IEEE Trans. Power Syst. 31(6), 4887–4900 (2016)
33. Medjroubi, W., Matke, C. & Kleinhans, D. : SciGRID - An Open Source Reference Model for the European Transmission Network (v0.2). https://www.scigrid.de (2015). Accessed 31 August 2021
34. Brown, T., Hörsch, J., Schlachtberger, D.: PyPSA: Python for power system analysis. J. Open Res. Softw. 6(1) 4, (2018)
35. Bayer, E.: Report on the German power system. Agora Energiewende, (2015)
36. Wang, X., Hickernell, F.J.: Randomized Halton sequences. Math. Comput. Model. 32(7-8), 887–899 (2000)

37. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. Adv. Neural Inf. Process. Syst. 14, 849–856 (2002)
38. Wang, X., Davidson, I.: Flexible constrained spectral clustering. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining 563–572 (2010)
39. Quirós-Tortós, J., Sánchez-García, R., Brodzki, J., Bialek, J., Terzija, V.: Constrained spectral clustering-based methodology for intentional controlled islanding of large-scale power systems. IET Gener. Transm. Distrib. 9(1), 31–42 (2015)
40. Morgenstern, O., von Neumann, J.: Theory of Games and Economic Behavior. Princeton, NJ: Princeton University Press, (1953)
41. Panteli, M., Mancarella, P., Trakas, D.N., Kyriakides, E., Hatziargyriou, N.D.: Metrics and quantification of operational and infrastructure resilience in power systems. IEEE Trans. Power Syst. 32(6), 4732–4742 (2017)
42. McRae, G.J., Tilden, J.W., Seinfel, J.H.: Global Sensitivity Analysis - A computational implementation of the Fourier amplitude sensitivity test (fast). Comput. Chem. Eng. 6(1), 15–25 (1982)
43. Cannavó, F.: Sensitivity analysis for volcanic source modeling quality assessment and model selection. Comput. Geosci. 44, 52–59 (2012)
44. Boser, B.E., Guyon, I.M. & Vapnik, V.N. : A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 1–9. New York, NY: ACM (1992)
45. Narsky, I., Porter, F.C.: Reducing multiclass to binary. Stat. Anal. Techn. Part. Phys. 1, 371–379 (2013)
46. Noebels, M., Preece, R., Panteli, M.: Training data for effect of preventive actions. Figshare (2021). http://doi.org/10.48420/14784570
47. Duan, K., Keerthi, S., Poo, A.: Evaluation of simple performance measures for tuning SVM hyper parameters. Neurocomputing 51, 41–59 (2001)

How to cite this article: Noebels, M., Preece, R., Panteli, M.: A machine learning approach for real-time selection of preventive actions improving power network resilience. IET Gener. Transm. Distrib. 16, 181–192 (2022). https://doi.org/10.1049/gtd2.12287