Fault Detection for agents on power grid topology optimization: A Comprehensive analysis

Malte Lehna [*]
Fraunhofer IEE, Kassel University
Kassel
Germany

Mohamed Hassouna[*]
Fraunhofer IEE, Kassel University
Kassel
Germany

Dmitry Degtyar
Fraunhofer IEE
Kassel
Germany

Sven Tomforde[†]
Kiel University: Intelligent Systems
Kiel
Germany

Christoph Scholz[‡]
Fraunhofer IEE, Kassel University
Kassel
Germany

Abstract

Optimizing the topology of transmission networks using Deep Reinforcement Learning (DRL) has increasingly come into focus. Various DRL agents have been proposed, which are mostly benchmarked on the Grid2Op environment from the Learning to Run a Power Network (L2RPN) challenges. The environments have many advantages with their realistic grid scenarios and underlying power flow backends. However, the interpretation of agent survival or failure is not always clear, as there are a variety of potential causes. In this work, we focus on the failures of the power grid simulation to identify patterns and detect them in advance. We collect the failed scenarios of three different agents on the WCCI 2022 L2RPN environment, totaling about 40k data points. By clustering, we are able to detect five distinct clusters, identifying common failure types. Further, we propose a multi-class prediction approach to detect failures beforehand and evaluate five different prediction models. Here, the Light Gradient-Boosting Machine (LightGBM) shows the best failure prediction performance, with an accuracy of 82%. It also accurately classifies whether a the grid survives or fails in 87% of cases. Finally, we provide a detailed feature importance analysis that identifies critical features and regions in the grid.

***K*eywords** Electricity Grids · Learning to Run a Power Network · Clustering · Forecasting · Deep Learning · Reinforcement Learning · Topology Optimization

1 Introduction

With the increase of renewables in the energy mix and the resulting variability, the optimization of power grids has become increasingly complex. One possible approach is topology optimization, i.e., switching buses at a substation level, which is a more cost-effective alternative to redispatching [22] and can improve grid stability [4]. However, identifying the correct topology options is computationally expensive [22]. One possible solution for the topology optimization could be DRL, as it can grasp large states and cope with the complexity [29].

Within the Machine Learning (ML) research community, the L2RPN challenges are the preferred research benchmarks [17, 24, 23, 22, 28]. The underlying Grid2Op package of the Transmission System Operator (TSO) RTE provides

[*] Both authors contributed equally.
malte.lehna@iee.fraunhofer.de, https://orcid.org/0000-0003-0621-1442 -
mohamed.hassouna@iee.fraunhofer.de, https://orcid.org/0000-0001-9927-1625

[†] https://orcid.org/0000-0002-5825-8915

[‡] https://orcid.org/0000-0002-8719-8261

several environments for DRL approaches.[4] Within the environments, researchers train their agents on various chronics, which are scenarios containing one week of power grid data, including generator injections, load consumption, and structural information like planned outages and unexpected hazards, to identify effective policies. To rate agents, a scoring function is provided [23], which depends on the agent's ability to maintain grid stability for a week in the test chronics. Several topology-based DRL agents have been applied to the L2RPN challenges of RTE in the last years [22]. However, it is not always clear why an agent survives or fails in a specific chronic, as the environments contain stochastic components that change the behavior on repeated runs. Therefore, we analyze different grid failures in the IEEE118 grid and provide a predictive approach to detect possible failures in advance.

2 Contribution

We take a closer look at agent failures to address the types of failures that can occur when using a topology-based agent. To do so, we aggregate the data of over 40k failed chronics from three different agents and cluster the failure types to identify specific patterns.

Next to clustering, we provide a multi-class prediction framework to identify whether an agent is currently in a situation where survival can be ensured, or if failure is imminent. Based on the observation of the grid state, we can detect whether the grid will survive or if a failure will occur in 5, 3, or 1 time-steps, corresponding to 25, 15, and 5 minutes respectively. We test different prediction models and are able to identify with 87% binary accuracy, whether grid failure is imminent. With a detailed feature importance analysis, we are further able to detect critical features and components in the electricity grid that are crucial for failure prediction. Overall, the contributions can be summarized as follows:

- We evaluate three agents from Ref. [19] with ten different seeds on the IEEE118 grid (WCCI L2RPN training environment). With 1662 chronics, we create a detailed dataset of around **40k grid failures**.
- We perform a **clustering analysis** that identifies distinct grid failure types and categorize them into five meaningful clusters, enhancing understanding of agents' shortcomings and their association with specific failure types.
- We build a **multi-class forecasting framework** to identify the risk of failure beforehand.
- We test five different forecasting models and identify the **LightGBM as the best performing candidate**.
- With an additional **feature importance** analysis, we are able to identify critical features and regions within the IEEE118 grid.

The rest of the paper is structured as follows. We provide an overview of the related work in Section 3 and our methodology in Section 4. We then present the experimental setup in Section 5, followed by the results in Section 6. Finally, we discuss our results in Section 7 and conclude in Section 8.

3 Related Work

With the L2RPN challenges, several approaches have been proposed to optimize the topology of transmission grids with DRL. In most cases, the DRL algorithms are combined with heuristic components [5, 11, 19, 20] to increase performance and reduce the action space. Other researchers prefer to use evolutionary algorithms with planning components [31] or Monte Carlo Tree Search (MCTS) [10]. Additionally, some approaches use Graph Neural Networks (GNNs) to represent the graph-structured state information [12]. However, most researchers report only the score and in some cases the survival of their agent on the test chronics. Understandably, they often do not show the shortcomings of their agents, making comparisons or even fault interpretation difficult. The most detailed analysis of the environments and agents comes from the L2RPN challenges themselves, e.g., in [23, 25]. They analyze the behavior of different agents and also show some examples using the Grid2Viz package.[5] Alternatively, [3] introduce Grid2Onto for Grid2Op environments. Their application is a recommender system based on a semantic model to display and select propositions from an agent for the respective grids.

With respect to fault prediction, previous studies have investigated electricity grids, as evidenced by the surveys for transmission grids [6] and distribution grids [8]. However, to the best of our knowledge, there is no specific work on fault prediction using DRL agents for topology optimization. The closest work was done by [26], who used an LSTM model to predict failures on the IEEE 35 bus network. However, they focus more on interlocking and cascading failures and not directly on the outcome of DRL agents. Therefore, there is still a need for further research in this area.

[4] Grid2Op: `https://grid2op.readthedocs.io/en/latest/` (last accessed 09/06/2024).

[5] Grid2Viz `https://github.com/rte-france/grid2viz` (last accessed 09/06/2024).

4 Methodology

4.1 Descriptive Analysis with Clustering

In the descriptive analysis, we want to identify different types of grid failures to better understand the shortcomings of the agents. However, the underlying data is high-dimensional and requires reduction. To ensure that we have less correlation in our clustering, we use the Principal Component Analysis (PCA) [1] with standardized variables. The dimensionality is reduced by transforming the data onto the coordinate system of the principal components, which are orthogonal to each other. We then select all major components that added at least 5% of the variance to the data, thus reducing the dimensions effectively. In our case, this yields six components with an explained variance of 85%.
With a reduced number of components, we combine the PCA with k-means clustering, similar to [9]. Introduced by [21], k-means clustering divides the data into k-clusters, where each data point is assigned to the nearest cluster centroid based on its distance. By iteratively assigning precise centroids, the distances within the cluster is minimized. Since the cluster size k has to be chosen, we iterate over different sizes and choose the cluster size based on the inertia and the silhouette metric [18].

4.2 Forecasting Models

In our forecasting framework, we go beyond a binary prediction of failure versus non-failure of the grid. Instead, we adopt a multi-class approach, forecasting potential failures at three distinct time steps before they occur, namely five, three and one time-steps ahead. This allows for a more granular understanding of the impending failures and provide insights into the decision-making of the models such as the average probability distribution among all classes. We employ the following five prediction models and compare their performance. In particular, we utilize gradient boosting techniques for their strong performance across various prediction tasks and their ability to easily provide insights into feature importance.

Random Forest (RF) [13] is an ensemble learning method that constructs simple decision trees during training and outputs the majority vote of their predictions for classification tasks. This approach enhances model accuracy and robustness by reducing overfitting and improving generalization compared to individual decision trees.

Extreme Gradient Boosting (XGBoost) [7] uses gradient boosting to iteratively build decision trees, such that each new tree is trained to correct the residual errors of all previous trees. It employs advanced techniques like regularization, weighted quantile sketch, and sparsity awareness to improve model accuracy, prevent overfitting, and handle large datasets efficiently.

LightGBM [16] is an ensemble learning method for classification that builds decision trees using a gradient boosting framework, optimizing for speed and memory efficiency. Unlike XGBoost, LightGBM uses a histogram-based algorithm and leaf-wise tree growth, which significantly enhances training speed and reduces memory usage, especially for large datasets.

Category Embedding Model (CEM) [14] is a basic yet effective feed-forward neural network. Its architecture integrates categorical features through a learnable embedding layer, making it a suitable starting point and baseline for comparison with other models.

GANDALF [15] stands for Gated Adaptive Network for Deep Automated Learning of Features and optimizes feature selection and engineering using Gated Feature Learning Units (GFLU). GFLU leverages a gating mechanism to iteratively refine feature representations and filter out noisy data, as well as utilizes stage-specific feature masks, allowing for hierarchical and adaptive feature selection. This results in a robust feature representation, which is then used by an Feed-forward Neural Network (FNN) to produce final predictions [15].

For the hyperparameter tuning of all models, Optuna [2] is employed to facilitate an efficient hyperparameter search. We utilize Tree-structured Parzen Estimator (TPE) [30] as our optimization method, as it maintains two probability distributions: one for good configurations and another for all sampled configurations. By focusing on promising regions of the hyperparameter space, TPE directs the search more efficiently than traditional methods like grid or random search.

5 Experimental Setting

5.1 Agents

In order to provide a comprehensive analysis of agent failures, it is essential to look at different agents. While there are numerous agents available for consideration, we decided to use the agents of [19] since their data was already available to us. The three agents in question are:

Do-Nothing Agent (*$DoNothing$*) The first agent does not interfere with the grid and only provides DoNothing actions. This agent is considered the baseline and we expect a lot more failures from it.

Senior Agent ($Senior_{95\%}$) This agent is based on the CurriculumAgent [20], which selects topology actions on individual substations, when the threshold of $\rho_{max,t} >= 0.95$ is breached. The agent further includes rule-based components, such as periodic reversion to the starting topology [19, 20] and line reconnection. The agent is more advanced, but changes the substations quite often in uncertain events.

Topology Agent ($TopoAgent_{85-95\%}$) The last agent is an extension of the $Senior_{95\%}$ and is described in detail in [19]. The main difference is that the agent searches for overall suitable Target Topologies (TTs) when the $\rho_{max,t}$ is in the interval of $0.85 < \rho_{topo} < 0.95$. In [19], the $TopoAgent_{85-95\%}$ was able to achieve better results, thus we expect differences to the $Senior_{95\%}$.

5.2 General Data Collection

To ensure a in-depth analysis, it is useful to look at a wide range of different environment scenarios. For this reason, we chose the WCCI 2022 L2RPN environment [27], with a total of 1662 chronics. These chronics are created with an expected electricity mix of 2050, i.e. with only 3% fossil fuels remaining in the electricity mix [27], so more variability is expected. Given the stochastic components of the environment, we run each chronic a total of 10 times with different master seeds for each agent.[6] Afterwards, the failed chronics were selected.

Cluster Data: For the clustering approach, we want to compare the observation of the failure with the initial situation of the environment. Thus, we collected the observation at the start of each chronic $obs_{t=0}$ and one time step before failure $obs_{t=n-1}$. Since each observation has 4295 variables, we aggregate the variables. First, since line capacity ρ is an important indicator of grid stability, we collect the maximum and average line capacity ρ_{max} and ρ_{mean} of the $obs_{t=n-1}$. Additionally, we counted the number of disconnected lines $\#lines_{dis}$ and the total $ts_{overflow}$, representing the cumulative duration during which the lines $l \in L$ have been in an overflown state ($\rho_l \geq 1.0$). Second, we account for the number of substation changes $\#sub_{changed}$, by comparing the switched substations to $obs_{t=0}$. Third, it is necessary to account for the different power flow changes in the grid. For the load consumption, the generator injections, and both sides of the lines (origin and extremity), we record the maximal active load value ($load^p_{max}, gen^p_{max}, line^{ex,p}_{max}, line^{or,p}_{max}$) and the average value ($load^p_{mean}, gen^p_{mean}, line^{ex,p}_{mean}, line^{or,p}_{mean}$) as a fraction to the first observation $obs_{t=0}$.[7] The fraction is calculated for the loads, generators, and lines. As an example, we show the calculation for the $load^p_{max}$ and $load^p_{mean}$ with γ being one load in the set of all loads $\gamma \in \Gamma$:

$$load^p_{max} = \max_{\gamma \in \Gamma} \left[load^p_{\gamma,t=n-1} / load^p_{\gamma,t=0} \right] - 1 \tag{1}$$

$$load^p_{mean} = \frac{1}{|\Gamma|} \sum_{\gamma=0}^{\Gamma} (load^p_{\gamma,t=n-1} / load^p_{\gamma,t=0}) - 1 \tag{2}$$

Next to the cluster variables, we also collected the survival times ($t_{survived}$) as well as other descriptive variables for our analysis.

Forecasting Data: For the forecast, we distinguish between failure and survival scenarios. For failures, we focus on observations at time steps $t = n - 1$, $t = n - 3$, and $t = n - 5$, with n being the failed time step. For survived observations, we sample from the chronics observations with a high $\rho_{max,t}$. However, to avoid overlapping, the $obs_{t=survived}$ must be at least $n - 6$ away from the failed time step. To avoid data leakage, we split the chronics

[6] To ensure that there was no cherry picking, we chose the ten seeds completely random using `np.seed`(8888). For each seed, we also calculated the `_statistics_l2rpn_dn` and `_statistics_l2rpn_no_overflow_reco$` separately.

[7] We excluded reactive power flow and voltage angle due to high correlation with active power flow or constant values. Elements with a zero in $obs_{t=0}$ were also excluded to avoid mathematical errors. See appendix table 4 for a summary of all variables.

beforehand into train, validation, and test sets, ensuring that no earlier data point from the test or validation set is found in the training set. This approach yielded 189 563 observations, with half of those surviving ($obs_{t=survived}$), and the remaining half is split evenly among $obs_{t=n-1}$, $obs_{t=n-3}$, and $obs_{t=n-5}$. Subsequently, through the sampling different chronics, the data was divided into a training set of size 152 234, a validation set of size 18 881, and a testing set of size 18 448. Thanks to the courtesy of RTE, the validation environment of the WCCI 2022 challenge was made available to us. Consequently, our findings can also be validated on an out-of-distribution (OOD) sample of size 6036.

5.3 Metrics

We evaluate the performance of our prediction models using several key metrics. First, we include the **Accuracy**, which is the ratio of correctly predicted instances to the total instances. This is the metric on which all models are trained. Further, with the multi-class classification problem, we also include the **Balanced Accuracy** to address the imbalance in the data sets. It is defined as the average of recall obtained for each class. The **F1 score**, which can be interpreted as a harmonic mean of precision and recall, reaches its best value at 1 and worst score at 0. For our specific use-case, we also want to measure the **Binary Accuracy**. This metric is defined by the accuracy of the one-vs-all model that categorizes all failures into a single class and survival into another. It effectively assessing the model's ability to distinguish between survival and failure observations.

6 Results

6.1 Clustering Results

Cluster Identification: With the aggregated clustering data, we perform PCA and select six principal components, describing 85% of the variance. Afterwards, we execute the k-means clustering and identify five clusters as the best cluster size for our data.[8] Subsequently, the clusters are grouped and the statistical components, including the mean, median, and quantiles of the variables, are analyzed to identify the distinctive characteristics of the clusters. We provide in Table 1 the mean values of all cluster variables. The following clusters were identified:

1. ***Changed Topology***: This cluster has the highest $\#sub_{changed}$ among all clusters.
2. ***Decreased Load Consumption***: In this cluster the distinct characteristic is the lower load consumption of $load^{p}_{mean}$ and $load^{p}_{max}$.
3. ***Disconnected Power Lines***: With the highest $ts_{overflow}$, $\#lines_{dis}$ and ρ_{max} this cluster is directly linked to the overload of the power lines.
4. ***Increased Generator Injections***: This cluster has by far the highest generator values of gen^{p}_{mean} and gen^{p}_{max}.
5. ***Increased Power Flow on Power Lines***: As the last cluster we identify the highest power flow in $line^{ex,p}_{max}, line^{or,p}_{max}, line^{ex,p}_{mean}$ and $line^{or,p}_{mean}$ as distinct feature.

Interestingly, we can see in Figure 1 that the clusters *Increased Generator Injections* and *Disconnected Power Lines* have distinct observations that can clearly be separated from the other clusters. In contrast, there seems to be multiple clusters with an increased $\#sub_{changed}$. With respect to Table 1, several noteworthy observations can be made. First, there appears to be a significant increase in generator output in the fourth cluster, with a factor of up to 125 times the gen^{p}_{mean}. Given that the relative change to $t = 0$ in active power is compared, it can be explained that some generators only produce a minor amount at $t = 0$ due to the fact that all scenarios start at nighttime. Secondly, it can be observed that the only instance of a decrease in the mean values can be found in the second cluster with $load^{p}_{mean}$. Given Kirchhoff's current law, i.e., the aggregated loads always have correspond to the aggregate generation in the whole grid, the lower $load^{p}_{mean}$ can only show localized imbalances in parts of the grid.

Cluster distribution across Agents: Following the general description of the clusters, we are interested in the distribution across the agents, as seen in Table 2. First, we see that *Changed Topology* and *Disconnected Power Lines* are quite large containing 39% and 30% of the data, while the *Increased Generator Injections* only has a total of 405 samples. With the χ^2 independence test ($\alpha = 0.05$), we tested the H_0-hypothesis that the clusters and agents are independent from each other. We could clearly reject the hypothesis with a $p - value$ of 0.0, which is a very strong indication that there is an association between clusters and agents. Comparing the clusters with the agents, we can see that a large portion of the $DoNothing$ failures are due to the disconnection of power lines (56%) as well as the reduced load consumption (26%). In contrast, the $Senior_{95\%}$ and $TopoAgent_{85-95\%}$ have most of their failures in the

[8] The plot of both inertia and silhouette can be found in the appendix figure 7.

Table 1: Mean values of the clustering variables, grouped by one of the five clusters. Based on multiple statistical characteristics, we name the clusters accordingly. We highlight the highest mean value of each variable.

Cluster	Changed Topology	Decreased Load Consumption	Disconnected Power Lines	Increased Generator Injections	Increased Power Flow on Power Lines
$ts_{overflow}$	5.32	6.89	**12.60**	10.32	9.96
$\#lines_{dis}$	0.79	1.58	**3.93**	1.37	2.12
$\#sub_{changed}$	**5.54**	1.71	0.92	2.94	3.57
ρ_{max}	1.03	1.13	**1.43**	1.25	1.24
ρ_{mean}	0.40	0.38	0.36	**0.41**	0.38
$load^{p}_{max}$	0.33	0.09	**0.35**	0.34	0.34
$load^{p}_{mean}$	0.12	-0.06	0.13	**0.14**	0.13
gen^{p}_{max}	4.78	2.69	3.66	**614.55**	2.31
gen^{p}_{mean}	0.27	0.10	0.08	**125.29**	-0.07
$line^{ex,p}_{max}$	0.51	0.21	0.60	0.58	**1.33**
$line^{ex,p}_{mean}$	19.16	12.24	22.07	19.56	**106.85**
$line^{or,p}_{max}$	0.51	0.20	0.61	0.58	**1.33**
$line^{or,p}_{mean}$	18.96	12.27	22.44	19.10	**108.58**

Table 2: Quantity and Frequency of failures in the clusters for the three agents. We denote the number of occurrences per cluster and the percentage within brackets. We highlight for each agent the most frequent cluster. Note that the amount of failure data across the agents differs.

	$DoNothing$	$Senior_{95\%}$	$TopoAgent_{85-95\%}$	Total
Changed Topology	1573 (10%)	**7191 (61%)**	**6676 (58%)**	15440 (39%)
Decreased Load Consumption	4345 (26%)	2239 (19%)	2508 (22%)	9092 (23%)
Disconnected Power Lines	**9225 (56%)**	1360 (12%)	1424 (12%)	12009 (30%)
Increased Generator Injections	178 (1%)	121 (1%)	106 (1%)	405(1%)
Increased Power Flow on Power Lines	1153 (7%)	784 (7%)	752 (7%)	2689 (7%)
Total # Obs	16.474	11.695	11.466	39.635

topology cluster. Here, the $TopoAgent_{85-95\%}$ has with 58% fewer topology failures, compared to the $Senior_{95\%}$'s 61%. Instead, the $TopoAgent_{85-95\%}$ has 3% more load consumption errors than the $Senior_{95\%}$.

Cluster Survival Time: Besides the distribution across the agents, we are also interested in the survival time. In Figure 2 we visualize the number of steps until failure with $t_{survived}$ and observe a clear difference between clusters. The *Increased Generator Injections* cluster fails incredibly fast, with a median survival time of only half a day (one day equals 288 steps). The median of the *Decreased Load Consumption* cluster is a little better. As contrast, we see that the *Changed Topology* cluster has the longest survival time with a median value of 496 steps, which might be related to the capabilities of the $TopoAgent_{85-95\%}$ and $Senior_{95\%}$. For the two remaining cluster, we have a relatively similar distribution, which is understandable since both focus on line results.

6.2 Forecasting Results

Quantitative forecasting results: In Table 3, we present the performance of all evaluated models with regard to the defined metrics of Section 5.3. LightGBM outperformed the other models across all metrics, achieving an accuracy of 82% and a high binary accuracy for predicting survival vs. failure of 87%. Interestingly, the performance remains consistent across both test and OOD data, with binary accuracy also holding steady at 87%, indicating no data leakage and robust generalization to novel scenarios. In second place, the other gradient boosting approach XGBoost achieved a relatively good performance, however, it was not able to reach that of LightGBM.

Compared to the gradient boosting approaches, the neural network-based methods CEM and Gated Adaptive Network for Deep Automated Learning of Features (GANDALF) perform very similarly and achieve an accuracy approximately

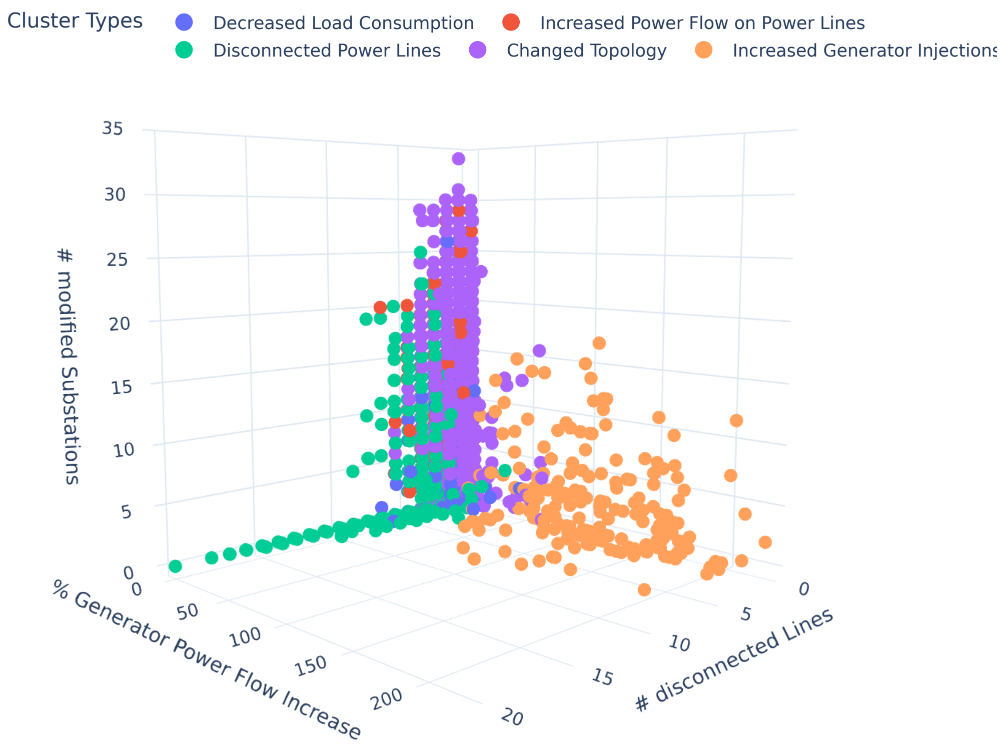

Figure 1: 3D visualization of the five clusters, where each point represents a failure of the agents. The axis in the plot are the gen^{p}_{mean} on the y-axis, $\#lines_{dis}$ on the x-axis and $\#sub_{changed}$ as z-axis. The points are colored according to their respective clusters.

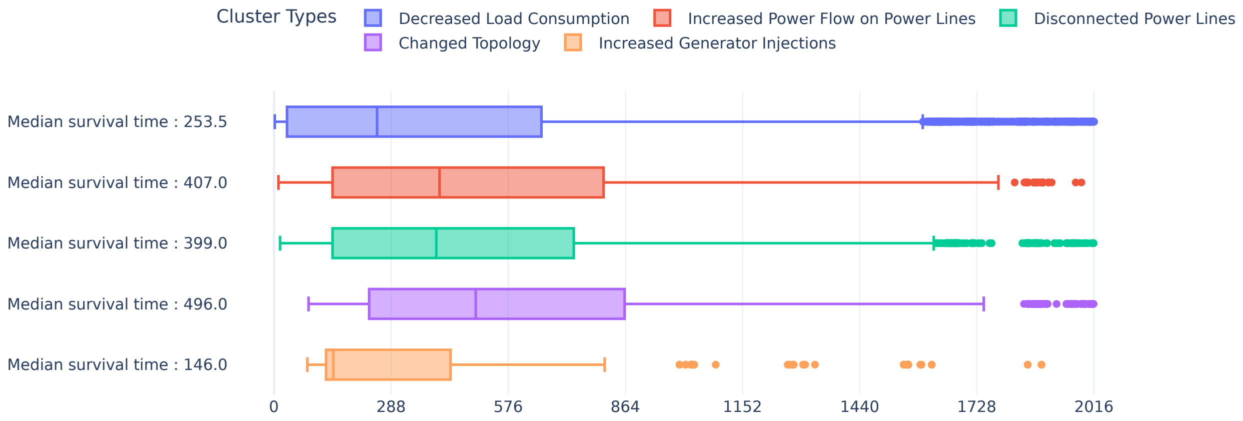

Figure 2: Box plot of the survival time $t_{survived}$ of each cluster. We further report the median survival time. The ticks on the x-axis correspond to a full day, e.g., 288 steps. Note that an agent must survive a total of 2016 time steps to complete the chronics successfully.

3% lower than LightGBM. Interestingly, both achieve a high accuracy for the binary problem of predicting survival vs. failure, outperforming XGBoost. It is also notable that both neural network-based methods achieve overall similar scores on both the test dataset and the OOD dataset. However, the scores are still lower than those achieved by LightGBM but comparable to XGBoost. Therefore, the GANDALF architecture did not achieve a higher performance than the CEM model, though it is more complex. In the last place is RF, which performed the worst across all metrics. With regards to the imbalanced nature of the data, it is important to also look at the balanced accuracy and the f1 scores. As expected,

Table 3: Forecasting results of the different models on the hold-out test data. The last two columns show the balanced accuracy and binary accuracy results on the OOD test data.

	accuracy	balanced accuracy	f1 micro	binary accuracy	OOD balanced accuracy	OOD binary accuracy
RF	0.73	0.62	0.73	0.82	0.61	0.82
XGBoost	0.80	0.73	0.80	0.83	0.73	0.83
LightGBM	**0.82**	**0.76**	**0.82**	**0.87**	**0.76**	**0.87**
CEM	0.79	0.73	0.79	0.84	0.73	0.84
GANDALF	0.79	0.72	0.79	0.84	0.72	0.83

we observe a decrease in balanced accuracy compared to the regular accuracy score. All models except RF exhibit a deviation of approximately 6%.

Qualitative Forecasting result: We take a closer look at the probability distribution of the best-performing model LightGBM. Figure 3 shows the averaged probability output of the model against the ground truth of all four classes. As we can see, the model selects the correct classification on average and is especially sure in the cases of survival, i.e., $obs_{survived}$ with 90%, and imminent failure, i.e., $obs_{t=n-1}$ with 81%. In the case of $obs_{t=n-5}$ we can see uncertainty in the model, since the model selects the surviving $obs_{t=survived}$ in 38% of the cases. This shows that it is difficult to distinguish $obs_{t=n-5}$ from a surviving observation, as it takes only a sudden change, such as an adversarial attack, to destabilize the grid very quickly. This can be observed for later timesteps as well, although to a lesser degree.

Feature Importance : As a last result, we examine the feature importance of the LightGBM model. For this we use the gain metric, which represents the improvement in accuracy brought by a feature to the branch. By evaluating the gain, we identify which features contribute the most to the model's predictions, allowing for a better understanding of the underlying patterns and relationships within the data. With respect to the features of the LightGBM, we look at the top 30 features, visualized in Figure 4. We group the features based on their type and distinguish among descriptive,

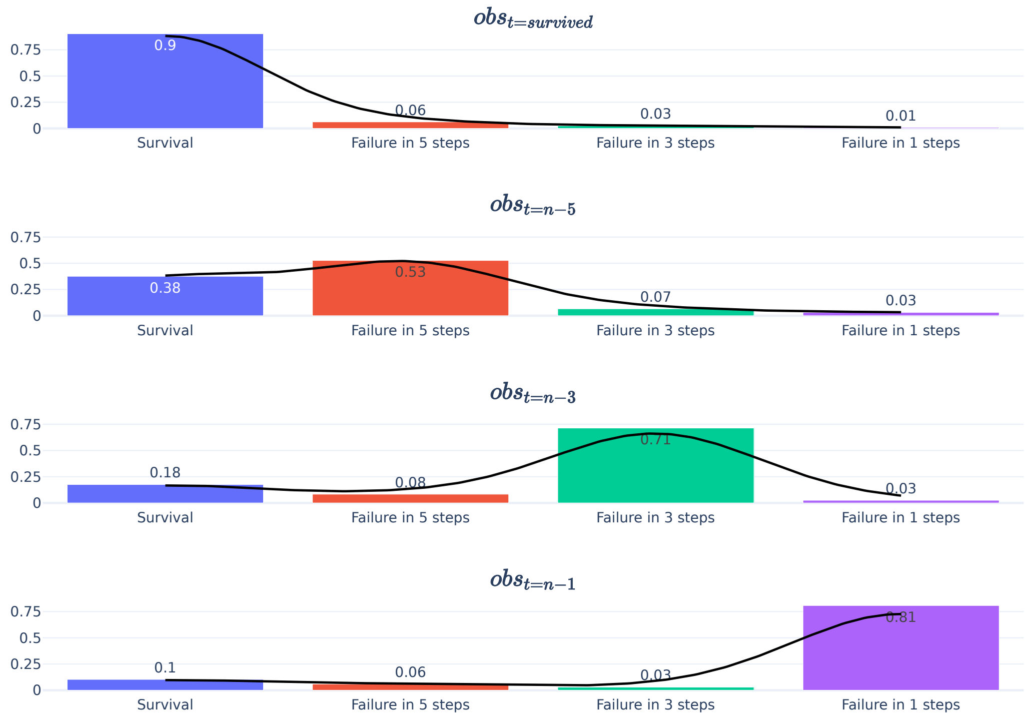

Figure 3: Average probability distribution of LightGBM for the ground truth of $obs_{t=survived}$, $obs_{t=n-5}$, $obs_{t=n-3}$ and $obs_{t=n-1}$. The probability output is averaged for all observations. Black lines visualizes the kernel density estimation across the classes.

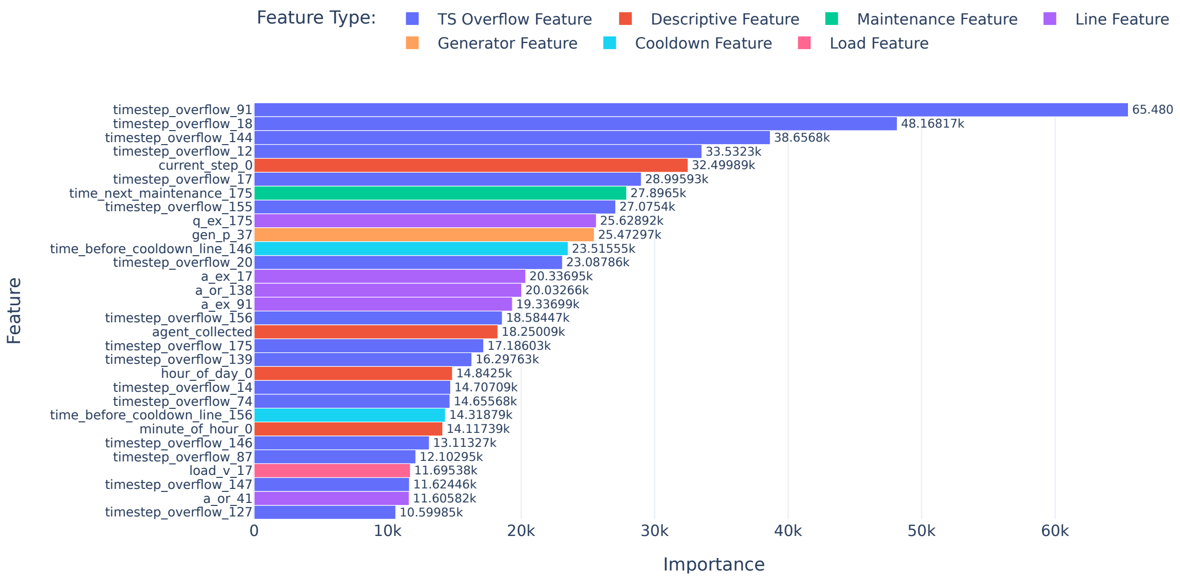

Figure 4: Feature importance of the 30 most important features according to the LightGBM model. The color indicates the type of variable.

generator, load, and line features (power flow, cooldown, $ts_{overflow}$, maintenance). Beginning with the interpretation of these results, we observe that 16 of the most important features are of type $ts_{overflow}$ which denotes the number of time steps since the power line has $\rho_l \geq 1.0$. This is not surprising, since overloaded lines are a strong indicator for an unstable grid. However, we also have three important descriptive features ($current\_step$, $minute\_of\_hour$, and $hour\_of\_day$). These suggest that the likelihood of power grid failures is influenced by temporal features, reflecting patterns that may correlate with daily operational cycles or varying demand. Another important descriptive feature is the agent type in 17th place, since there is a clear difference in behavior between the $TopoAgent_{85-95\%}$, $Senior_{95\%}$ and the $DoNothing$ in terms of survival. It is plausible that some grid states would lead to a failure without remedial action, but an agent applying topological fixes could prevent this. Regarding the loads and generators, we can see that the first generator feature appears at position 10, while the first load feature appears only at position 27. This is quite interesting, considering that the decreased load cluster is relatively frequent. It is possible that the grid's stability is more dependent on the fluctuating nature of largely weather-dependent renewable generators than on loads. Examining the overall picture, it is unsurprising that the feature importance is dominated by line features. Moreover, many features correspond to the same lines. For instance, three distinct features corresponding to line 175 appear among the top 30 importance values alone. To simplify the interpretation of feature importance and to better understand the broader impacts of different types of grid components, we aggregated the feature importance values by averaging the importance values of related features. The results of the aggregation are visualized in Figure 5, where we display the ten most important lines, generators and loads of the grid.

In the first region (A) we have two loads and a line of high importance. By looking at the failures we could identify the line between substations 21 and 22 as reason, as it is often targeted by the adversarial agent. This attack is a certain failure because the demand of the loads cannot be changed and the only generator that provides electricity is in substation 18. This causes an overflow in all the lines between substations 18 and 21, leading to a cascading failure in less than 3 steps. Thus, by disconnecting the line, the adversarial agent causes a failure, which to our surprise is very common with 4057 out of 39635 cases.

Region (B) highlights the connection between two sub-grids: the line between substations 68 and 76 connects the upper right sub-grid to a major substation in the bottom right sub-grid. The most important generator 37 connects to substation 68 and frequently injects up to 500% of its starting power. Furthermore, substation 76 is connected to the central substation 79 with two lines, indicating an increased power flow and overall significance of this connection. Accordingly, these lines are deemed as very important by the prediction model. Both substations are very important since all power flow to the majority of this sub-grid have to be routed through them, hence they serve as the entry point to this part of the grid. Additionally, two other paths (80-67-64 or 80-67-68) from substation 79 to the upper right sub-grid have to traverse substation 80, hence the connection between 79 and 80 is also considered very important by the model. Overall, region (B) can be characterized as a high line power flow region, having 3 important generators (37, 42 and 43) in the vicinity, connecting two sub-grids, and being the only entry point to the bottom right area of the grid.

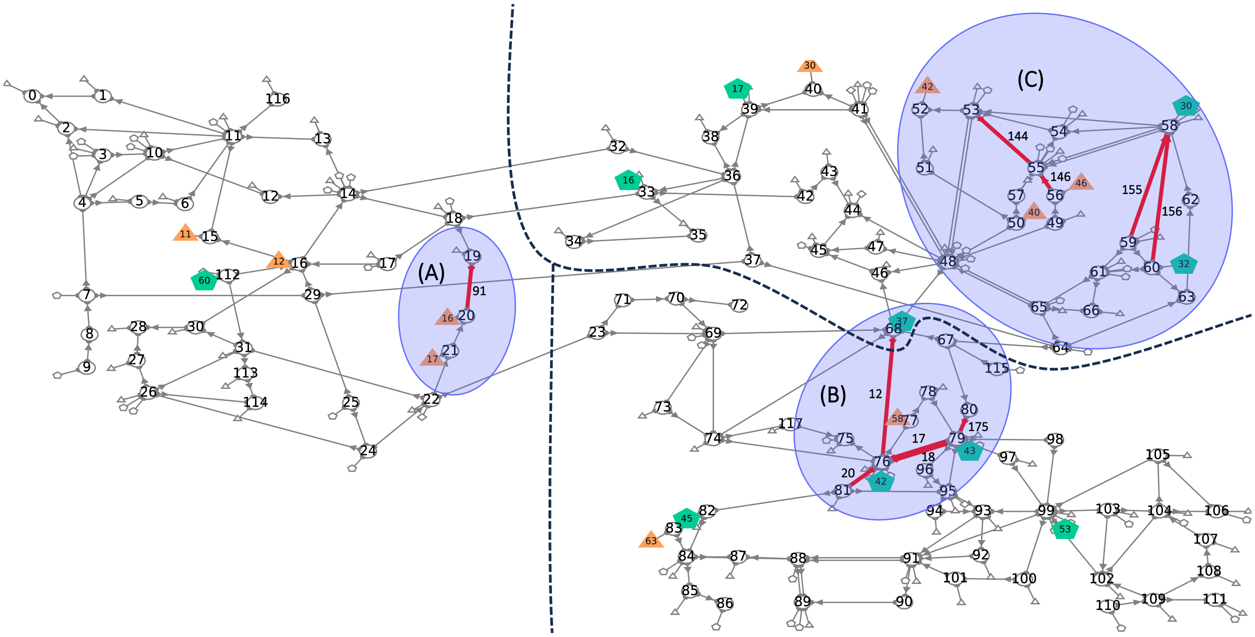

Figure 5: Top 10 important lines (red), generators (green), and loads (yellow) for failure prediction. The numbers on the marked lines correspond to the line ids of Figure 4. Sub-grids are separated with dotted lines. For better clarity, we grayed out the less important elements of the grid and highlighted 3 significant regions (A,B,C) of important grid features.

Region (C) marks a section of the grid where frequent line attacks by the adversarial agent occur. In fact, 10 out of 23 possible line attacks happen in this sub-grid with 5 line attacks in region (C) alone. Both the lines connecting substations 58 to 62 and 62 to 63 are part of the attacked lines, leading to a higher load on the other lines connecting the group of substations $(59, 60, 61, 63, 64, 65, 66)$ in the lower area of this sub-grid. Namely, these are the lines connecting substation 58 to 59 and 60, respectively, as correctly identified by the importance of the model. Moreover, three out of the ten most important loads are present in this region, connected at substations 50, 52, and 56, respectively. In order to supply these loads, a high load and importance at the lines connecting these loads to bigger substations is to be expected. This is exacerbated by the fact that the lines connecting substation 48 to 50 and substation 48 to 53 can be attacked, leaving less possible paths to these loads available.

7 Discussion

Looking at the results, we can see that the clustering was quite successful. We were able to identify at least two of the five clusters that are highly correlated with the agents, confirming our expectations. Moreover, both advanced agents were able to significantly reduce the number of failures due to line disconnection. Since the $TopoAgent_{85-95\%}$ has slightly less errors due to topology changes, we can also confirm [19]. In terms of the remaining clusters, the results are not as clear. For the line and generator clusters, we have very similar cluster sizes across all agents. For the load cluster, we could at least observe a slight reduction for the advanced agents. Overall, the results indicate that there is still a lot of optimization potential, requiring additional redispatching optimization. With the feature importance, we were also able to detect the design error in region (A) of Fig. 5, showing a possible requirement to incorporate load shedding in future action spaces of Grid2Op. Agents should focus on preventing the disconnection of this line due to overflows at all cost by performing suitable remedial actions in a timely manner. With respect to our predictive models, we were able to achieve an accuracy of 82%, showing that model correctly detect possible failures in the gird in most cases. More importantly, the binary accuracy of 87% showed, that we are able to correctly detect whether the agent is struggling in the grid. This makes the real-world applicability more interesting, as we can alert the operator in time, with $obs_{t=n-5}$ corresponding to 25min before failure. As future work, improving the performance of the failure prediction could be achieved by utilizing GNNs to capture the inherent graph structure present in the power grid topology. Further, a continuous analysis should be conducted that includes redispatching components to target the reduced load cluster by actively adjusting redispatch and curtailment. Furthermore, it might be interesting to focus on the specific regions of the grid that received high feature importance, as they seem to be responsible for the performance of the agents. This could be achieved by specifically training the DRL agent on these regions, or even a hierarchical approach. Including topological actions that specifically target these regions into the action space might bring a boost in performance as well. Finally, the prediction of failures has a potential in guiding decisions on when a DRL agent should take remedial actions based on them, rather than fixed thresholds of maximum line capacity $\rho_{max,t}$.

8 Conclusion

In this paper, we provide a detailed failure analysis on the WCCI L2RPN environment, which has not been done to this extent yet. We examined the failures in the grid of three different agents across ten seeds, resulting in 40k data points. Our analysis is twofold, with a detailed cluster analysis and a prediction of the failures in advance. For the cluster analysis, we were able to identify five specific clusters that showed distinct causes for the failures. The failures exhibited varying survival times and could be attributed to some extend to specific agents. As a second part, we propose a multi-class forecasting approach to detect the failures ahead of time. On the data we tested five different model types and found the LightGBM to be the most suitable model for the prediction. The feature importance afterwards revealed critical regions of the grid that could be specifically targeted in future work.

Acknowledgement

This work was supported by (i) the research group Reinforcement Learning for Cognitive Energy Systems (RL4CES) founded by the German Federal Ministry of Education and Research (01|S22063B), (ii) Graph Neural Networks for Grid Control (GNN4GC) founded by the Federal Ministry for Economic Affairs and Climate Action Germany(03EI6117A) and (iii) AI4REALNET founded by the European Union's Horizon Europe Research and Innovation program under the Grant Agreement No 101119527. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Competing Interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Abdi, H., Williams, L.J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics (2010)

[2] Akiba, T., Sano, S., et al.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)

[3] Amdouni, E., Khouadjia, M., et al.: Grid2onto: An application ontology for knowledge capitalisation to assist power grid operators. In: International Conference On Formal Ontology in Information Systems-Ontology showcases and Demos (2023)

[4] Bacher, R., Glavitsch, H.: Network topology optimization with security constraints. IEEE Transactions on Power Systems (1986)

[5] Chauhan, A., Baranwal, M., et al.: Powrl: A reinforcement learning framework for robust management of power networks. arXiv preprint arXiv:2212.02397 (2022)

[6] Chen, K., Huang, C., et al.: Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. High voltage (2016)

[7] 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 (2016)

[8] Dashti, R., Daisy, M., et al.: A survey of fault prediction and location methods in electrical energy distribution networks. Measurement (2021)

[9] Ding, C., He, X.: K-means clustering via principal component analysis. In: Proceedings of the twenty-first international conference on Machine learning (2004)

[10] Dorfer, M., Fuxjäger, A.R., et al.: Power grid congestion management via topology optimization with alphazero. arXiv preprint arXiv:2211.05612 (2022)

[11] EI Innovation Lab, Huawei Cloud, Huawei Technologies: NeurIPS Competition 2020: Learning to Run a Power Network (L2RPN) - Robustness Track. `https://github.com/AsprinChina/L2RPN_NIPS_2020_a_PPO_Solution` (2020), accessed 22-Jan-2023 on Github

[12] Hassouna, M., Holzhüter, C., Lytaev, P., Thomas, J., Sick, B., Scholz, C.: Graph reinforcement learning for power grids: A comprehensive survey (2024), `https://arxiv.org/abs/2407.04522`

[13] Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. IEEE (1995)

[14] Joseph, M.: Pytorch tabular: A framework for deep learning with tabular data (2021)

[15] Joseph, M., Raj, H.: Gandalf: Gated adaptive network for deep automated learning of features (2024)

[16] Ke, G., Meng, Q., et al.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems (2017)

[17] Kelly, A., O'Sullivan, A., et al.: Reinforcement learning for electricity network operation. arXiv preprint arXiv:2003.07339 (2020)

[18] Kodinariya, T.M., Makwana, P.R., et al.: Review on determining number of cluster in k-means clustering. International Journal (2013)

[19] Lehna, M., Holzhüter, C., et al.: Hugo–highlighting unseen grid options: Combining deep reinforcement learning with a heuristic target topology approach. arXiv preprint arXiv:2405.00629 (2024)

[20] Lehna, M., Viebahn, J., et al.: Managing power grids through topology actions: A comparative study between advanced rule-based and reinforcement learning agents. Energy and AI (2023)

[21] MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (1967)

[22] Marot, A., Donnot, B., et al.: Learning to run a power network challenge for training topology controllers. Electric Power Systems Research (2020a)

[23] Marot, A., Donnot, B., et al.: Learning to run a power network challenge: a retrospective analysis. In: NeurIPS 2020 Competition and Demonstration Track. PMLR (2021)

[24] Marot, A., Donnot, B., et al.: Learning to run a power network with trust. Electric Power Systems Research (2022)

[25] Marot, A., Guyon, I., et al.: L2rpn: Learning to run a power network in a sustainable world neurips2020 challenge design (2020b)

[26] Mu, Z., Xu, P., et al.: Cascading fault early warning and location method of transmission networks based on wide area time-series power system state. IEEE Journal of Radio Frequency Identification (2022)

[27] Serré, G., Boguslawski, E., et al.: Reinforcement learning for energies of the future and carbon neutrality: a challenge design. arXiv preprint arXiv:2207.10330 (2022)

[28] Subramanian, M., Viebahn, J., et al.: Exploring grid topology reconfiguration using a simple deep reinforcement learning approach. In: 2021 IEEE Madrid PowerTech (2021)

[29] Viebahn, J., Naglic, M., et al.: Potential and challenges of AI-powered decision support for short-term system operations. CIGRE 2022 Paris Session (2022)

[30] Watanabe, S.: Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance (2023)

[31] Zhou, B., Zeng, H., et al.: Action set based policy optimization for safe power grid management. In: Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021 (2021)

9 Appendix

9.1 Cluster Variables

We summarize the distribution of the cluster variables in the following Table 4.

Name	Explanation	mean	std	min	25%	50%	75%	max
$ts_{overflow}$	Cumulative overflow time	8.25	6.78	0.0	3.0	6.0	12.0	71.0
$\#lines_{dis}$	Number of disconnected lines	2.02	2.27	0.0	0.0	1.0	3.0	20.0
$\#sub_{changed}$	Number of changed Substations	3.10	4.63	0.0	0.0	1.0	4.0	34.0
ρ_{max}	Max line capacity	1.19	0.28	0.61	1.03	1.11	1.33	2.0
ρ_{mean}	Mean line capacity	0.38	0.06	0.19	0.33	0.38	0.42	0.63
$load^{p}_{max}$	Max active load value of consumption	0.28	0.14	-0.14	0.22	0.30	0.37	0.88
$load^{p}_{mean}$	Mean active load value of consumption	0.08	0.10	-0.26	0.04	0.11	0.15	0.32
gen^{p}_{max}	Max active production value of genreators	10.03	70.03	-0.48	0.79	1.67	3.27	1573.0
gen^{p}_{mean}	Mean active production value of genreators	1.43	13.40	-0.98	-0.24	-0.0	0.30	224.02
$line^{ex,p}_{max}$	Max active power flow at line extremity	0.52	0.39	-0.22	0.24	0.46	0.73	4.60
$line^{ex,p}_{mean}$	Mean active power flow at line extremity	24.41	32.13	0.24	8.60	15.89	27.25	696.11
$line^{or,p}_{max}$	Max active power flow at line origin	0.53	0.39	-0.22	0.24	0.46	0.74	5.64
$line^{or,p}_{mean}$	Mean active power flow at line origin	24.56	33.86	0.23	8.57	15.84	27.15	730.49
$t_{survived}$	Survival Times	539.87	470.42	2.0	147.0	407.0	804.0	2016.0

Table 4: Variables for clustering. Note that $t_{survived}$ is not included in the clustering and is just reported to showcase the distribution.

Correlation of cluster variables: When looking at the correlation plot in Figure 6, there are already three interesting things to note. First, we see a high correlation in the power flow values between the maximum and average components, which is relatively self-explanatory. Second, we can see that the load values are somewhat correlated to $line^{ex,p}_{max}, line^{or,p}_{max}$. Interestingly, the generator values are not correlated with the line power flow in any way. Third, we see a negative correlation between $\#sub_{changed}$ and $\#lines_{dis}$, which can be explained by the fact that both the $Senior_{95\%}$ and $TopoAgent_{85-95\%}$ agents have components that automatically reconnect lines. However, $\#lines_{dis}$ is also negatively correlated with ρ_{max}, which is counter-intuitive. A possible explanation could be the Grid2Op rule that disconnects a line after three consecutive time steps of overload. When the line is disconnected, the highest ρ_{max} is no longer available, explaining the negative correlation. This explanation would primarily hold for consecutive observations, so it is quite interesting that it seems to work across multiple scenarios.

9.2 Inertia and Silhouette plot

The plot below (Fig. 7) illustrates the inertia values for different cluster sizes in our k-means clustering analysis, showcasing the selection process for determining the optimal number of clusters based on the elbow method and the corresponding decrease in inertia.

9.3 Additional Information

Error Statement We collected the error statement of every failed scenario to see, whether we could detect a pattern. The following three error statements were outputted by the underlying Grid2Op solver `lightsim2grid`:

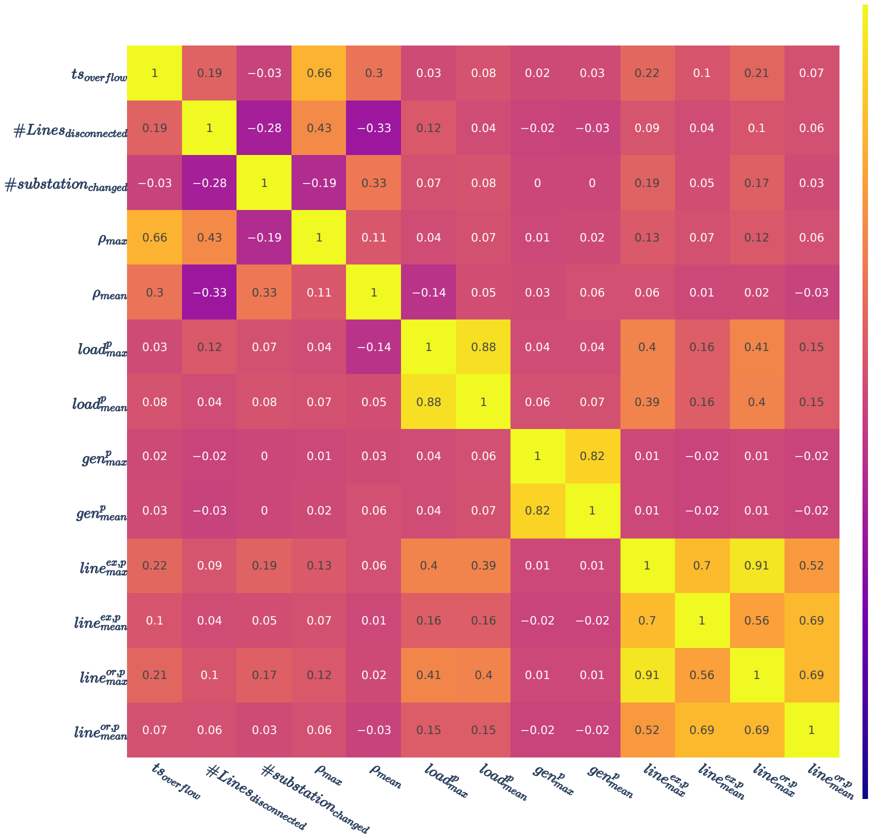

Figure 6: Correlation plot of the different cluster variables. The color range depicts the correlation values that go from -1.0 (blue) to 1.0 (yellow).

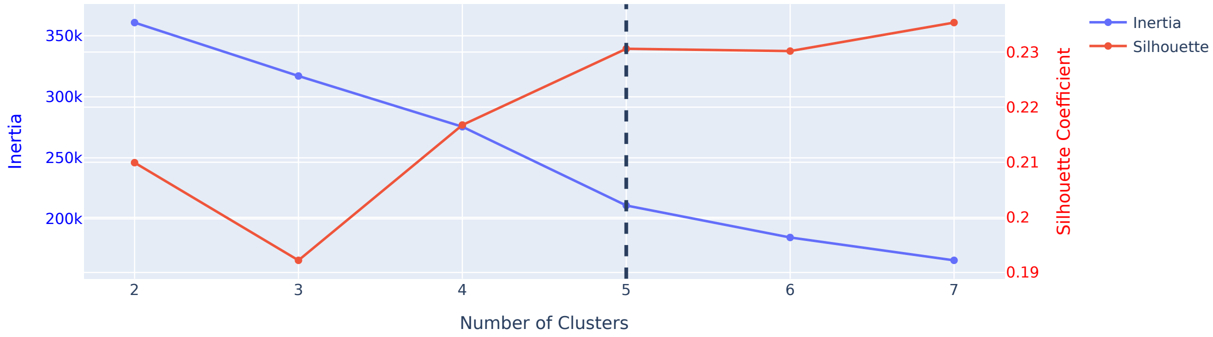

Figure 7: Inertia and Silhouette plot of different cluster sizes. As one can see, the inertia has a "elbow" at a cluster size of five. Similar, the silhouette shows one of its highest values at a size of five as well. Thus, we select a cluster size of five.

1. `Grid2OpException Divergingpower flow "Divergence of DC power flow (non connected grid) at the initialization of AC power flow. Detailed error: ErrorType.SolverFactor"`

2. `Grid2OpException Divergingpower flow "Divergence of AC power flow. Detailed error: ErrorType.TooManyIterations"`

3. `Grid2OpException Divergingpower flow "Divergence of DC power flow (non connected grid) at the initialization of AC power flow.  Detailed error: ErrorType.SolverSolve"`

We collected all error messages and mapped them to the clusters of Section 4.1. The results can be found in Table 5. While there is some difference in the percentage change, we can not find a distinct relation between cluster and error message.

Error Type	Changed Topology	Decreased Load Consumption	Disconnected Power Lines	Increased Generator Injections	Increased power flow on Power Lines
1	61.2%	63.97%	69.27%	68.15%	65.01%
2	27.47%	27.03%	24.71%	26.42%	26.18%
3	11.32%	9%	6.01	5.43%	8.81%

Table 5: In this Table, we map the three error types with the different clusters. For each cluster we aggregate the error types to percentage values for better viability.

Specific Descriptive Scenarios We further accounted for three descriptive variables for the scenarios:

- $Attack_{93}$: This variables indicates, whether the Line 93 is disconnected or will be disconnected in the next step
- This variable indicates, whether the grid was in its original state, i.e., all substation had their buses set to one and all lines were disconnected
- Lastly, we collected the percentage change of the generator 37, as this generator was directly at substation 68.

For the first two variables, we have the following occurence:

Specific Descriptive Scenarios

	$DoNothing$	$Senior_{95\%}$	$TopoAgent_{85-95\%}$	Total
$Attack_{93}$	1367	1312	1378	4057
$Stable$	169	817	646	1632

Table 6: Specific failure cases identified in descriptive analysis.

In case of the generator 37, we can see the following percentage increase across the scenarios:

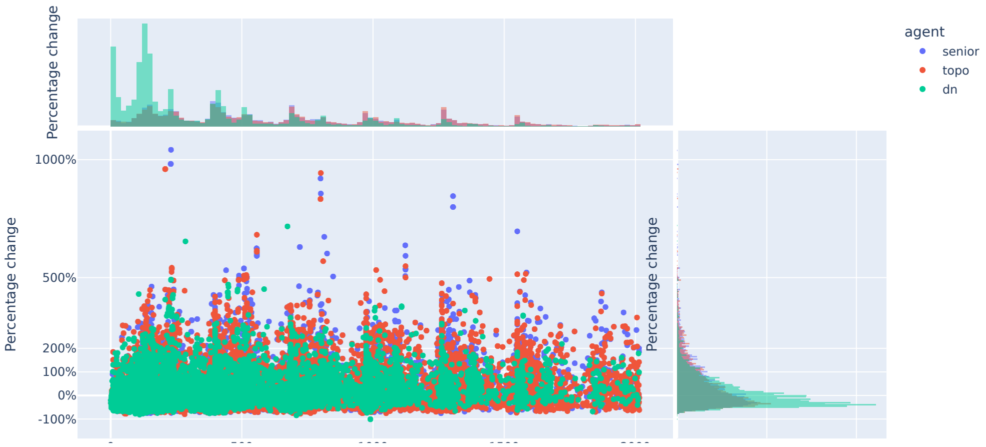

Figure 8: Visualization of the percentage change on the generator 37 in comparison to its $obs_{t=0}$.