Learning Schemes for Power System Planning and Control

Carter Lassetter
Oregon State University
lassettc@oregonstate.edu

Eduardo Cotilla-Sanchez
Oregon State University
ecs@oregonstate.edu

Jinsub Kim
Oregon State University
kimjinsu@oregonstate.edu

Abstract

In this paper, learning algorithms are leveraged to advance power system protection. Advancements in power system protection have come in different forms such as the development of new control strategies and the introduction of a new system architecture such as a microgrid. In this paper, we propose two learning schemes to make accurate predictions and optimal decisions related to power system protection and microgrid control. First, we present a neural network approach to learn a classifier that can predict stable reconnection timings for an islanded sub-network. Second, we present a learning-based control scheme for power system protection based on the policy rollout. In the proposed scheme, we incorporate online simulation using the commercial PSS/e simulator. Optimal decisions are obtained in real time to prevent cascading failures as well as maximize the load served. We validate our methods with the dynamics simulator and test cases RTS-96 and Poland.

1. Introduction

As power systems evolve over time, there is great desire to increase the robustness and reliability of energy delivery. It is increasingly important to ensure cascading failures of networks are limited. Such failures, such as the blackout of 2003 in North America [1], are well documented and could have been prevented with smarter controls [2]. Much work on protective relaying has been performed to mitigate the impacts of contingencies [3]. Normal discrete components make up an underlying protection scheme in a network that rely on a stream of measurements to operate. These relays trip when their associated thresholds are exceeded. Strict operation with discrete components may hinder protection schemes due to potential vulnerabilities in the grid that lay dormant until a certain contingency exposes an underlying problem. These problems can lead to an increased probability of cascading [4]. Further control of a system can be performed by network operators that have the ability to perform 'expert actions.' Many strategies focus on load shedding in attempts to bring voltage levels back to a tolerable range to avoid voltage collapse [5]. Other methods attempt to break the network up into self-sufficient islands to mitigate any propagating failures [6]. Emergency islanding was seen in the Europe blackout in 2006 in which a single overhead line trip caused the continent to divide into three main islands [7].

Networks are normally designed with redundancy to ensure that the loss of a single component will not have a significant impact on network operation, this is known as $N-1$ security. Further redundancies may be built in a system to improve the robustness of the grid, however may not be economically feasible. Due to this, $N-2$ security is not necessarily ensured for a given network. With the difficulty of obtaining this level of security, the ability to recover from said events with intelligent control schemes becomes necessary.

Impacts of outages need not be stressed, however the ability to guard against full scale blackouts whilst maximizing the amount of load served is desirable. From this, we focus on two main avenues when defending against cascading failures; the first includes direct improvement of network control in the form of action selection. This takes the form of preventative actions available to a network operator, such as direct load shedding or islanding. When exploring actions, off-line based approaches have yielded success in the past in the form of policy-switching [8]. Other methods that leverage online based solutions, such as policy rollout, have shown notable improvements on maximizing end load served whilst guarding against blackouts [9]. Policy rollout is an online solution that explores a certain action set and chooses the best action at a given time [10]. The work seen in [9] makes use of the COSMIC simulator [11] when leveraging policy rollout; we use another dynamics simulator, Siemens PTI PSS/e, to compare the results.

It is clear that directly improving control schemes

URI: http://hdl.handle.net/10125/50222
ISBN: 978-0-9981331-1-9

on the network as a whole is a possible solution when defending against cascading failures. The usage of intelligent actions may prove to both maximize network operation and mitigate full scale network outages. Another potential solution to cascading includes the creation of a more modular network. The ability to safely disconnect bad pieces of the network to isolate problems may prevent cascading failures. This second avenue to be explored needs further refinement on the control side as well.

These modular sub-networks are called microgrids which are comprised of load, generation and possibly energy storage [12]. Their main benefit is that they are self-sufficient and capable of operating independently of main grid support. Microgrids continue to be seen as a solution to current problems in our grid and are even being deployed by utilities [13]. With sub-networks having the ability to potentially separate from the main grid, it is important that we can observe important aspects of their operating points. With the explosion of new monitoring techniques, the ability to coordinate these networks becomes more feasible. PMUs allow for direct measurement of bus voltages/angles allowing better monitoring and control techniques to be developed [14]. Said measurements occur at high sample rates allowing new control techniques to operate in near real time [15, 16]. The implementation of microgrids along with new monitoring and control will greatly benefit power delivery reliability, mitigate potential outages, and aid in advancement to a smarter grid [17]. Unfortunately the integration of microgrids remains difficult due to the lack of control techniques during off-nominal operation.

Previous works in literature focus on reconnection of small microgrids consisting of one point of common coupling (PCC). Said techniques work with manual synchronization between the buses connected at the PCC in terms of voltages, angles, and frequencies [18, 19, 20]. The ability for smooth reconnection from synchronization is sought after by implementing load shedding, generator curtailment, as well as energy storage [21, 22]. With respect to larger microgrids with multiple PCCs, it becomes difficult to ensure synchronization at all points despite the obvious problem of needing manual changes to the network. A clear solution for multiple PCC synchronization is not known and may take much computation time to determine the stability of reconnection in a time sensitive situation. In addition, sub-optimal measurement locations create problems pertaining regular threshold solutions.

Due to the difficulties associated with creating a verbatim rule for reconnection other avenues must be explored [23]. With the advancements in Artificial Intelligence, we make use of neural networks to aid in producing a solution to classifying stability in a large domain of unexplored states. Many power system solutions have made use of Artificial Intelligence successfully [24, 25, 26, 27].

We draw upon previous work that predicts the stability of network reconnection with limited PMU measurements using Support Vector Machines (SVMs) [28]. To improve upon this work we use a neural network when building a classifier. The benefit of a neural network over SVM is mainly seen in easier training. A neural network is straightforward to build whereas SVMs may require difficult pre-processing potentially involving kernel functions. A neural network would prove to be an easier form of implementation of solving said problem. With the proposed technique, we make use of PMU measurements as an input to our neural network used to predict if the current state would lead to a stable or unstable reconnection. It is important to note that PMUs are not fully prevalent in power system monitoring at the moment, thus full observability of the system cannot be assumed. The proposed scheme needs to be built in a way such that it accounts for limited PMU measurements [29]. The usage of electrical distance may be used when selecting certain PMU locations [30, 31]. Due to this, we pick PMUs to be located near the PCCs in the test network used.

PMUs make use of GPS synchronization [32] which may open up attack platforms for adversaries. A well planned cyber attack may be designed to hijack a waveform by changing or shifting time synchronization which may degrade our classifier's performance. An incorrect decision at a key point in time may trigger cascading problems throughout the network which remain hidden until undiscovered by failures [3]. To guard against this, we make use of trustworthy subsets of PMUs in the face of potentially compromised PMUs.

The remainder of this paper is organized as follows. Section 2 briefly formulates the problem. Section 3 introduces the simulator used. Section 4 discusses the architecture for our proposed solutions. Section 5 shows the results and Section 6 provides the conclusions.

2. Problem Formulation

We separate our problem into two main tasks. The first focuses on solving the problem of coordinating a reconnection between a microgrid and main grid. The second looks into directly improving network control in the face of contingencies.

2.1. Microgrid Reconnection

In this section we formulate the prediction for stability of a microgrid and main grid reconnection in terms of a learning problem. We propose to make use of real time data measurements in the form of PMUs to build our classifier. Said measurements will be used as the features in our learning problem. When dynamically predicting the potential stability of a reconnection, we make use of the current time step of measurements to create a real-time estimator. The class label will correspond to the stability of reconnection at a given time step. As a result, we will have two classes consisting of 'stable' and 'unstable.'

When learning a classifier, we make use of a set of training data consisting of a number of input feature vectors $x_1, ..., x_n$ and their associated class labels $y_1, ..., y_n$. Each feature vector along with the corresponding target will be used in our neural network to learn weights. An output softmax layer is utilized to associate 'scores' to each class in relation to the input feature vector. The loss will be based on how close we were to our true target class and back-propagate through the network to modify the weights for increased future performance. Upon completion of training, it should be possible to feed in a new unseen feature vector to the neural network and estimate the associated class label. These unseen feature vectors represent the potential unknown state of our network corresponding to a new combination of voltages/angles at different observable buses.

2.2. Network Operation Improvement

We also look into a solution that focuses on directly controlling the network as a whole in attempts to minimize the impacts of cascading failures. In the face of network contingencies, we make use of a policy based solution to aid in making decisions in real time.

Policy rollout is an attractive solution to network control due to it's ability to be performed online. When tasked with choosing an action, policy rollout may make use of a model and transition function to explore an action set and ultimately choose the best action to its knowledge. Depending on the model and transition function, an action in a given state may result in a new state with a given probability. This is best represented with equation (1) depicting the transition from state s at time t to state s^* at time $t+1$.

$$T(s_t, a_t, s^*_{t+1}) = P(s^*_{t+1}|s_t, a_t) \qquad (1)$$

One can overcome the issue of probabilistic transitions by running several simulations in a monte-carlo fashion to obtain the average results of performing a particular action. When a model does not have stochasticity, the transition becomes deterministic and eliminates the need to explore a given action at a state multiple times. We make the assumption that our network is deterministic which drastically reduces the time complexity of exploring our action/state space.

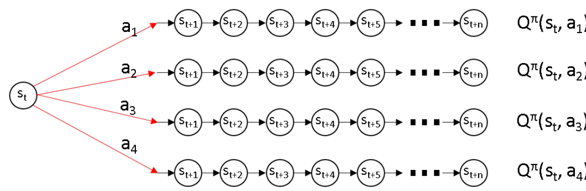

Figure 1. Policy rollout with depth one search.

A control policy is implemented based on acting greedily according to an estimated action-value function $\tilde{Q}^{\pi}$ of a rollout policy π. This action-value function refers to 'how good' an action is at a given state. As shown in Figure 1, four actions exist. A depth one search allows each action to be explored with a baseline policy being implemented afterwards. As an example, we explore each of the four possible actions at s_t, at each time step thereafter we perform the action 'Do nothing.' This 'Do nothing' policy means that no external action is taken by network operators, however the underlying protection scheme will still operate. We can then act greedily by selecting the best action in accordance to equation (2).

$$\pi_*(s) =_{a \in A} \tilde{Q}^{\pi}(s, a) \qquad (2)$$

It is important to note that $\tilde{Q}^{\pi}(s, a)$ is the estimated action-value for a particular action in a unique state. To clarify, the states in Figure 1 for each action may be different.

The estimated action-value function, $\tilde{Q}^{\pi}(s, a)$, can be found by taking a monte carlo approach. As a result, we would perform the exploration seen in Figure 1 many times for each action (necessary if non-deterministic). We will have many potential trajectories based on our action selection, this can be denoted as $\tau = s_0 a_0 s_1 a_1 \ldots s_{H-1} a_{H-1} s_H$. One can see that different trajectories can be made up of a different string of state and action sequences. If we denote the total reward of trajectory τ as $R(\tau)$, it will contain each reward r from each state in the trajectory seen in equation (3). Future rewards may be weighted less by using the discount factor $\beta \in [0, 1]$.

$$R(\tau) = \sum_{i=0}^{H} \beta^i r(s_i) \tag{3}$$

If we are tasked with estimating $\tilde{Q}^{\pi}(s,a)$, we may create many trajectories and sample m of them [33]:

$$\tilde{Q}^{\pi}_m(s,a) = \frac{1}{n_m(a)} \sum_{i=1}^{n_m(a)} R(\tau_i^a) \tag{4}$$

In this case, we find the action-value function for state s while taking action a. The total reward of the i^{th} sampled trajectory when taking action a is denoted by $R(\tau_i^a)$. We label the amount of times action a was taken in m samples as $n_m(a)$. After obtaining sufficient samples, we may choose the best action in accordance to equation (2).

3. Introducing the Dynamics Simulator

For both problems at hand, it is necessary to use a dynamics simulator to both acquire data and perform simulations on networks. We make use of the commercial simulator PSS/e to perform the needed tasks.

3.1. Microgrid Reconnection

We first build a dynamic simulation to allow data acquisition. We made use of the Poland test case consisting of 2383 buses. The Poland case is made up of 5 zones, we use zones 1-4 to represent the main grid while zone 5 represents the microgrid. The microgrid is innately similar in form to the rest of the network, however the response between the two networks should maintain adequate consistency with a fully detailed and developed microgrid representation [34]. Dynamics are implemented on said case which consist of salient machines for the generators, IEEE Type 1 exciters, and IEEE Type 2 governors. We further this case by building a protective scheme consisting of over-current line relays, undervoltage/underfrequency load shedding relays, and overfrequency/underfrequency machine trip relays. The addition of dynamics and protection settings allow an adequate representation of a real world network.

Upon completion of building the steady state and dynamic Poland network, it is important to create different load distributions throughout the case to ensure diversity among our data. We randomly shuffle our load data throughout the case along with scaling said load. Load scaling for both active and reactive power is done and shown by Eqs. (5) and (6):

$$P_{new} = P_{old} + \theta P_{old}, \quad \theta \sim\ U(-a, b) \tag{5}$$

$$Q_{new} = Q_{old} + \gamma Q_{old}, \quad \gamma \sim\ U(-a, b) \tag{6}$$

We build 24 different operating points with the aforementioned method. Upon acquiring these operating points, we further diversify our network case to create new initial conditions by re-implementing load scaling at a smaller scale than creating new operating points. This gives us over 200 different configurations of our first built network. We then run our dynamic simulation on each newly created network. The dynamics simulator perform network evaluations every 1/120 seconds which will represent our time step. Our dyamics simulation is performed by islanding our makeshift microgrid (zone 5) from the main grid (zones 1-4) at 2 seconds. We specify a reconnection window between 20-40 seconds in which we randomly reconnect the microgrid and main grid. We perform roughly 60 dynamic simulations with random reconnection times for each network created. For each simulation, we save voltages/angles of each PMU for all time points before reconnection. We make use of the single time point before reconnection for our stability classifier.

3.1.1 Data Features and Labels/Targets. As previously stated, we generate our output dynamics files from different network configurations. Voltages and angles are embedded within the output file after simulation and thus a parser is developed to obtain the data before reconnection. We specify a set of PMUs that are available to populate our feature set. The PMUs that lie on the microgrid side are: 127, 171, 165, 126, 186, 166, 174, 167, 178, 2218, 2124, 2249, 2331, 2226, 2234 whereas the PMUs that exist on the main grid are: 10, 15, 214, 225, 303, 315, 335, 118, 125, 139, 1607, 1761, 128, 140, 141. The entire set of PMUs may be used or a smaller subset when creating our stability predictor. We make use of the time step immediately before reconnection for our stability predictor. The class label is set to 'stable' if the network converges after reconnection and has at least 2370 buses connected and in service. If the previous criteria are not met, the case is labeled 'unstable.'

3.2. Network Operation Improvement

We work with the RTS-96 test case which is comprised of three identical networks connected to one another. Dynamic models are included on the generators in the form of the 'GENSAL' salient generator model, IEEE type 1 exciter and IEEE type 2 governor. The other components, include buses, transmission lines, transformers, and loads. For simulating dynamics we implement a time step of $\frac{1}{120}$ seconds specifying how often the network case state is reevaluated.

Basic protection is implemented in the form of overcurrent relays on branches and under voltage/frequency relays on buses. The discrete protective elements are implemented to alleviate local stress within the network and protect components. The addition of this protection scheme will show the impacts of different contingencies and potential cascading failures.

The state of the network is comprised of many different elements that evolve over time. The simulator keeps track of these as state variables which include dynamic variables that change at each time step. The topology of the network is remembered as well which consists of the status of components such as: lines, transformers, loads, and generators. In addition the attached protective relays and their state are saved at each time point as well.

4. Implementing a Learning Scheme

After laying the groundwork with our simulator, we now propose the solutions pertaining to each problem.

4.1. Microgrid Reconnection

With regards to the Microgrid Reconnection problem, we make use of the neural network to learn how to separate stable vs. unstable cases.

4.1.1. Neural Network Training. Upon running our simulations, we may now define our data sets and train our networks. We make use of a basic Neural Network to create a classifier.

We build a set of examples that consist of PMU voltage and angle measurements near the interconnection point and their associated class (stable/unstable). We separate our full set of examples into two distinct sets, one to train the network, and the other to test it. Our train set normally will not have an equal distribution of classes, thus we use the common oversampling technique to avoid poor training [35]. It is important to address the issue of scalability in our network. We may come across our network that is operating in an unknown operating point that we have yet to see/train for. To prove our method can defend against ill conditioned situations, we train on 18 of the created distinct operating points and test on the other 6. We create different random distributions of train/test operating sets to show that results are not confined to a well separated train/test set of said operating points. We explore the potential of utilizing smaller subsets from our available set of PMUs,this is achieved by utilizing only available features when training/testing a new network. After training our classifier from a certain set of PMUs we test on the test set and compare the ground truth class and the predicted class. Accuracy is recorded for both classes for a reliable accuracy measurement.

Several configurations of a neural network were built to test our approach, however a single layer with a sigmoid or tanh activation function proved to work the best. We utilized different numbers of hidden units to test the network as well and settled on 100. The sigmoid activation function for the hidden layer was connected to a softmax layer to create the classifier. It is important to address that the main modifications to a neural network include: Number of hidden nodes, hidden layers, activations functions, output loss function. The problem at hand is not limited to a unique setup of the aforementioned modifications which implies the ability to setup an architecture is not difficult and sub-optimal architectures do not have a large adverse impact.

4.2. Network Operation Improvement

With regards to the Network Operation Improvement problem, we leverage a policy rollout based approach to improve network operation.

4.2.1. Application to Network Operation. We use the algorithm policy rollout in tandem with the dynamics simulator PSS/e to demonstrate an improved approach to prevent cascading failures in a network as well as increase load survivability. We focus on the RTS-96 case with a predefined protection scheme. We attempt to improve upon the operation of discrete protective devices and expert based actions.

4.2.2. Baseline Polcies. We leverage similar baseline policies shown in [9]. These include: Shedding global load and isolating zones in which contingencies occur. Due to the difficulty with PSS/e interaction of state variables and the necessity of adding user defined models, we did not make use of the ‘hysteretic load shed‘ or HLS baseline policy.

A key thing to note about these policies is that both Isolate and ShedGlobal occur with short delay after the associated contingencies. Similarly, the same delay is implemented with the policy rollout approach to ensure no bias occurrs. The remaining protective elements within the system will continue to operate until the end of simulation. We make the assumption that the ability to shed load and disconnect certain branches is available which in reality may be limited. In practice,

Table 1. Unseen operating point case accuracies for Poland network with subsets of PMUs

PMU location [bus number]	Class 1 accuracy	Class 0 accuracy
118, 127, 166, 2249	98.0%	93.3%
139, 165, 2218, 2226	95.5%	90.9%
118, 141, 166, 174, 1607, 2218, 2249, 2331	96.2%	93.5%
118, 139, 167, 178, 214, 315, 335, 2226	96.2%	93.6%
15, 118, 125, 126, 139, 140, 167, 171, 174, 2124, 2218, 2234	93.8%	97.0%
10, 118, 140, 214, 225, 303, 315, 335, 2124, 2218, 2226, 2234	93.3%	95.5%
10, 125, 126, 139, 167, 178, 186, 225, 315, 1607, 2124, 2218, 2226, 2234, 2249, 2331	96.9%	94.4%
118, 125, 126, 139, 165, 171, 174, 186, 214, 303, 335, 1607, 1761, 2218, 2226, 2249	96.9%	94.2%

load shedding or branch disconnection is performed by opening a circuit breaker. These devices may not always be located in the necessary configuration, however similar performance should occur.

4.2.3. Available Actions for Policy Rollout. The available actions within our network include both load shedding and islanding. Due to policy rollout being an online method, it is important that the amount of actions is not so great that exploration becomes infeasible. If we allowed load shedding at all available locations concurrently we would come to an action space of the size $O(2^b)$. Due to this we use three expert actions that are also drawn from [9]:

> **ShedZone**(z, p): Shed a proportion of $p \in [0, 1]$ of all loads within zone z.
>
> **ShedGloabl**(p): Shed a proportion of $p \in [0, 1]$ of all loads within the network.
>
> **Island**(z): Island zone z from all other zones in the network.

This action space abstraction becomes more necessary as the network scales. Computing power also may impact how an action space is chosen as more power may correspond to the ability to make less abstract actions. Similarly, the ability to search the action space deeper or longer is impacted by available processing power.

5. Results

5.1. Microgrid Reconnection

For the Poland test case we built 24 different operating points and different initial conditions for each. As a result, we obtained roughly over 200 unique networks stemming from the original one. We ran around 60 dynamic simulations on each network to obtain over 10,000 examples for our stability predictor. We implemented a neural network to achieve our goal and made use of several subsets of PMUs to prove that a small set of PMUs may still be used to acquire high confidence when making predictions. Table 1 shows the accuracies between Class 1 (stable) and Class 0 (unstable) for different configurations of PMUs. It is important to note that it is possible to achieve high accuracies even with a reduced set of PMUs boding well for future applications with limited PMU placement. As stated previously, the network was trained on a limited set of operating points and tested on a group of unknown operating points. The high accuracy observed on unknown operating points is a welcome result as it addresses the concern of making predictions on large networks with many unique states difficult to train for. It is also worth mentioning that the accuracies shown correspond to equal weights associated to each class meaning neither class is given preference over the other during training. This shows the most general form of the classifier in which the operator may want to reconnect the subnetwork as soon as possible, however is still weary of instability.

5.2. Network Operation Improvement

When simulating the RTS-96 case, we do not account for stochasticity. We leverage the baseline policies to get a sense of how well our network can survive during certain $N-2$ contingencies with expert actions. For all possible $N-2$ contingencies, we first perform the simulation with no expert control. We then take a little over 400 of the worst performing cases, with respect to load survival, and perform the given policies to demonstrate the increased performance. The global load shed action sheds 10% of the load at all shunts and the zone isolation works by disconnecting any tie line that connects a bus to a zone in which a contingency occurs. We also allow no expert action to take place and let only the protection scheme on the network operate. It is important to note that this baseline protection scheme exists for all policies. When testing the policy rollout algorithm we made use of the available actions: ShedZone(z,0.1), ShedGlobal(0.1), and Island(z) which allows shedding 10% of all shunts in the network, or in a given zone, as well as islanding any zone.

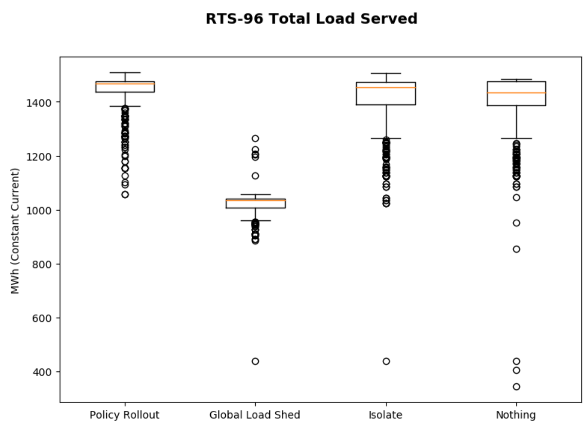

Figure 2. RTS-96 total load survivability with different policies.

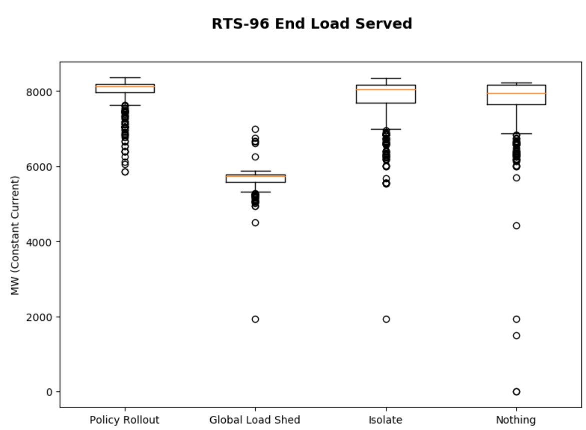

Figure 3. RTS-96 end load survivability with different policies.

When referencing Figs. 2, 3 one should note that the box represents the interquartile range (IQR), bounded by the first and third quartile. Anything outside of $1.5{*}$IQR, shown by the whiskers, is labeled as an outlier. In Figure 2 we see the total survivability of the RTS-96 case. This means we account for the loading at each time point and add up the total amount of load served over the entire duration of the simulation. Conversely, we look only at the end load served in Figure 3 to account for how much of the case has survived to the end. Both results from either metric look similar.

An interesting result to observe in Figs. 2, 3 is that it is possible to perform no expert actions and still obtain good network operation. This relies heavily on how well the protection in the scheme is configured. As seen, the protection scheme allows the case to survive many $N-2$ contingencies. The GlobalShed policy performs worst as it seems to shed unnecessary amounts of load to protect the case. The ShedZone performs relatively well, most likely due to the configuration of the RTS-96 case. Depending on the locations of contingencies, it is possible that the baseline protection automatically separates the network into the best islands. The policy rollout case seems to perform the best in which it can allow a higher average surviving load. To further discuss this point, the outliers seen in policy rollout are much more acceptable than in the other policies; for instance, when only the underlying protection scheme is used we can observe many cases that perform substantially worse than policy rollout.

With the cases evaluated, it can be seen that the policy rollout approach does seem to perform better that the other potential policies. Further evaluation is necessary for more extreme contingencies. The implementation of stochasticity is also important to check in the future as it will have an impact on the amount of time necessary to evaluate actions.

6. Conclusion

This paper introduces two main solutions to cascading failures in power systems. We discuss a novel approach to solving the problem of reconnecting microgrids. With the rollout of microgrids, it is necessary that control schemes guard against poor reconnection of microgrids. It is apparent the benefits that microgrids will contribute in future electrical power systems. This paper focuses, in part, on increasing the resiliency in their operation. It is shown that accuracies of around 90% can be obtained when reconnecting a microgrid, even in states of unknown operation.

We strengthen our contribution by exploring smarter controls on a network for blackout mitigation. Policy rollout was performed to demonstrate better performance when networks are posed with contingencies. The method can be performed online to explore a set of possible actions to ensure the best series of control actions are performed. Blackouts were mitigated with this technique, however we also demonstrate increased network performance pertaining to load served.

7. Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780.

8. Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions ofauthors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] J. Chadwick, "How a smarter grid could have prevented the 2003 U.S. cascading blackout," in *Power and Energy Conference at Illinois (PECI), 2013 IEEE*, pp. 65–71, Feb 2013.

[2] J. Chadwick, "How a smarter grid could have prevented the 2003 U.S. cascading blackout," in *Power and Energy Conference at Illinois (PECI), 2013 IEEE*, pp. 65–71, Feb 2013.

[3] S. Tamronglak, S. Horowitz, A. Phadke, and J. Thorp, "Anatomy of power system blackouts: preventive relaying strategies," *IEEE Transactions on Power Delivery*, vol. 11, no. 2, pp. 708–715, 1996.

[4] S. Soltan, D. Mazauric, and G. Zussman, "Cascading failures in power grids: Analysis and algorithms," in *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy '14, (New York, NY, USA), pp. 195–206, ACM, 2014.

[5] N. N. A. Bakar, M. Y. Hassan, M. F. Sulaima, M. N. M. Nasir, and A. Khamis, "Microgrid and load shedding scheme during islanded mode: A review," *Renewable and Sustainable Energy Reviews*, vol. 71, pp. 161 – 169, 2017.

[6] M. Golari, N. Fan, and J. Wang, "Two-stage stochastic optimal islanding operations under severe multiple contingencies in power grids," *Electric Power Systems Research*, vol. 114, pp. 68 – 77, 2014.

[7] C. Li, Y. Sun, and X. Chen, "Analysis of the blackout in europe on november 4, 2006," in *2007 International Power Engineering Conference (IPEC 2007)*, pp. 939–944, Dec 2007.

[8] R. Meier, E. Cotilla-Snchez, and A. Fern, "A policy switching approach to consolidating load shedding and islanding protection schemes," in *2014 Power Systems Computation Conference*, pp. 1–7, Aug 2014.

[9] J. Hostetler, *Monte Carlo Tree Search with Fixed and Adaptive Abstractions*. PhD thesis, Oregon State University, 2017.

[10] D. P. Bertsekas and D. A. Castanon, "Rollout algorithms for stochastic scheduling problems," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 2, pp. 2143–2148 vol.2, Dec 1998.

[11] J. Song, E. Cotilla-Sanchez, G. Ghanavati, and P. D. H. Hines, "Dynamic modeling of cascading failure in power systems," *IEEE Transactions on Power Systems*, vol. 31, pp. 2085–2095, May 2016.

[12] N. Lidula and A. Rajapakse, "Microgrids research: A review of experimental microgrids and test systems," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 186 – 202, 2011.

[13] T. W. Overton, "Oncor's system operating services facility, lancaster, texas.," *Power*, vol. 159, no. 8, pp. 48 – 51, 2015.

[14] A. Phadke, "Synchronized phasor measurements in power systems," *IEEE Computer Applications in Power*, vol. 6, no. 2, pp. 10–15, 1993.

[15] J. Burnett, R.O., M. Butts, and P. Sterlina, "Power system applications for phasor measurement units," *IEEE Computer Applications in Power*, vol. 7, no. 1, pp. 8–13, 1994.

[16] T. Cease and B. Feldhaus, "Real-time monitoring of the TVA power system," *IEEE Computer Applications in Power*, vol. 7, pp. 47–51, July 1994.

[17] N. Hatziargyriou, *Microgrids: Architectures and Control*. Wiley - IEEE, Wiley, 2013.

[18] F. Tang, J. M. Guerrero, J. C. Vasquez, D. Wu, and L. Meng, "Distributed active synchronization strategy for microgrid seamless reconnection to the grid under unbalance and harmonic distortion," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2757–2769, 2015.

[19] E. Alegria, T. Brown, E. Minear, and R. H. Lasseter, "Certs microgrid demonstration with large-scale energy storage and renewable generation," *IEEE Transactions on Smart Grid*, vol. 5, pp. 937–943, March 2014.

[20] M. A. Sofla and R. King, "Control method for multi-microgrid systems in smart grid environmentstability, optimization and smart demand participation," *IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–5, Jan 2012.

[21] S. K. Khadem, M. Basu, and M. F. Conlon, "Intelligent islanding and seamless reconnection technique for microgrid with upqc," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 3, pp. 483–492, June 2015.

[22] T. M. L. Assis and G. N. Taranto, "Automatic reconnection from intentional islanding based on remote sensing of voltage and frequency signals," *IEEE Transactions on Smart Grid*, vol. 3, pp. 1877–1884, Dec 2012.

[23] M. A. Sofla and R. King, "Control method for multi-microgrid systems in smart grid environmentstability, optimization and smart demand participation," *IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–5, Jan 2012.

[24] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector machines: a study on eunite competition 2001," *Power Systems, IEEE Transactions on*, vol. 19, pp. 1821–1830, Nov 2004.

[25] S. Kalyani and K. Swarup, "Classification and assessment of power system security using multiclass svm," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, pp. 753–758, Sept 2011.

[26] I. Saeh and M.W.Mustafa, "Machine learning classifiers for steady state security evaluation in power system," *International Journal of Computer Science*, vol. 9, no. 2, pp. 262–269, 2012.

[27] L. Wehenkel, "Machine learning approaches to power-system security assessment," *IEEE Expert*, vol. 12, pp. 60–72, Sep 1997.

[28] C. Lassetter, E. Cotilla-Sanchez, and J. Kim, "A learning scheme for microgrid reconnection," *IEEE Transactions on Power Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[29] M. Zhou, V. Centeno, J. Thorp, and A. Phadke, "An alternative for including phasor measurements in state estimators," *Power Systems, IEEE Transactions on*, vol. 21, pp. 1930–1937, Nov 2006.

[30] E. Cotilla-Sanchez, P. Hines, C. Barrows, and S. Blumsack, "Comparing the topological and electrical structure of the north american electric power infrastructure," *Systems Journal, IEEE*, vol. 6, pp. 616–626, Dec 2012.

[31] E. Cotilla-Sanchez, P. Hines, C. Barrows, S. Blumsack, and M. Patel, "Multi-attribute partitioning of power networks based on electrical distance," *Power Systems, IEEE Transactions on*, vol. 28, pp. 4979–4987, Nov 2013.

[32] K. Jones, A. Pal, and J. Thorp, "Methodology for performing synchrophasor data conditioning and validation," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1121–1130, 2015.

[33] H. Pham, *Continuous-time Stochastic Control and Optimization with Financial Applications*. Springer Publishing Company, Incorporated, 1st ed., 2009.

[34] J. Bialek, E. Ciapessoni, D. Cirio, E. Cotilla-Sanchez, C. Dent, I. Dobson, P. Henneaux, P. Hines, J. Jardim, S. Miller, M. Panteli, M. Papic, A. Pitto, J. Quiros-Tortos, and D. Wu, "Benchmarking and validation of cascading failure analysis tools," *IEEE Transactions on Power Systems*, vol. 31, pp. 4887–4900, Nov 2016.

[35] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, (New York, NY, USA), pp. 935–942, ACM, 2007.