Techniques for Improving Filters in Power Grid Contingency Analysis

Robert Adolf, David Haglin, Mahantesh Halappanavar, Yousu Chen, and Zhenyu Huang

Pacific Northwest National Laboratory, Richland, Washington, USA,
(Robert.Adolf@pnl.gov, David.Haglin@pnl.gov,
Mahantesh.Halappanavar@pnl.gov, Yousu.Chen@pnl.gov and
Zhenyu.Huang@pnl.gov)

Abstract. Electrical power grid contingency analysis aims to understand the impact of potential component failures and assess a system's capability to tolerate them. The computational resources needed to explore all potential x-component failures, for modest sizes of $x > 1$, is not feasible due to the combinatorial explosion of cases to consider. A common approach for addressing the large workload is to select the most severe x-component failures to explore (a process we call filtering). It is important to assess the efficacy of a filter; in particular, it is necessary to understand the likelihood that a potentially severe case is filtered out. A framework for assessing the quality/performance of a filter is proposed. This framework is generalized to support resource-aware filters and multiple evaluation criteria.

Keywords: Contingency Analysis, Contingency Selection, Resource-Aware Filtering, Multi-criteria Optimization

1 Introduction

In large-scale power transmission systems, predicting faults and preemptively taking corrective action to avoid them is essential to prevent rolling blackouts. Power grid contingency analysis is the study of the impact of potential component failures, which is frequently done with computational simulation. In the early part of this decade, the state of the art was $N-1$ contingency analysis, referring to the ability to predict the behavior of the electrical grid in response to a single random failure. However, in response to increasing load, dynamic power generation, and several wide-area blackouts in the early 2000's, the power engineering community has begun to realize the necessity of predicting the consequences of more simultaneous faults. This problem, generally termed "$N-x$ contingency analysis", grows combinatorially with the number of components in the grid, and for large power systems spanning thousands of components, the requirements to enumerate and simulate all possible scenarios rapidly becomes infeasible. Still, analyzing concurrent failure modes helps operators better understand a system's overall behavior, especially its response to cascading faults, and remains an important objective in the field.

One commonly-accepted method for avoiding this computational explosion is to use an approximation technique to estimate which elements in the grid are most likely to cause severe instability in the grid if they fail, and simulate only the failure of those pieces. This allows for devoting limited computational resources to the cases most likely to be important to the power grid. In this context, we see three avenues for general improvement:

1. Many filtering techniques exist in published literature [5], but in general, there is no universally agreed-upon methodology for evaluating the success of any particular one over another;
2. Although most filtering algorithms do reduce the number of simulated cases that need to be computed, there is no general approach to quantifying the degree to which the accuracy of a filter is affected by the reduction factor;
3. While it is true that there is a correlation between the severity of a single fault and the severity of subsequent faults, predicting the former is not sufficient to predict the latter in general. Thus, when solving the problem of $N - x$ contingency analysis, the set of sub-problems $N - 1$, $N - 2$, etc. actually compete for simulation resources and must be prioritized according to operational requirements; no documented technique exists to do so.

When viewed at a high level, contingency analysis is not altogether dissimilar from many existing large-scale data mining problems, and several of the unsolved issues that contingency analysis faces have analogs in other domains. In this paper, we present a cohesive solution for addressing all three of the deficiencies above with the hope of providing a jumping-off point for integrating the contingency analysis and data mining fields. More precisely, our contribution is an abstract framework that implements: a method for evaluating the efficacy of a filtering algorithm for contingency selection (Section 3), a technique for resource-aware integration of multiple filters (Section 4), and a strategy for managing competing heterogenous selection pipelines with a shared, constrained resource (Section 5).

2 Context

To a power engineer, contingency analysis is a complex, multi-disciplinary problem which often requires not only technical ingenuity, but also political and logistical savvy. Vast networks of embedded sensors and data aggregation nodes report parameters describing the state of a power grid upstream to control centers where the data is fed to massive numerical models. Operators then enumerate a host of possible failure conditions and simulate the grid's response, judging whether the possible consequences merit preemptive action. Managing this wave of distributed sensor data, building accurate electrical simulations, and assessing the relative benefits of corrective adjustments are all well outside the scope of this paper and rightly deserve entire conferences of their own accord.

Fundamentally, however, the computational workflow for the core problem can be radically simplified to look more familiar to the computer science community. The power engineer is essentially concerned with one question: *which* *failures are the most likely to cause severe damage to the grid?* From this perspective, we note that contingency analysis looks very similar to a scoring or ranking problem. The input to our framework is the set of all possible contingencies, each representing the possibility of a component failure somewhere in the transmission network. Each contingency is simulated and assigned a severity score by a vetted method, then handed off to an operator. Graphically, this process looks something like the diagram in Figure 1.

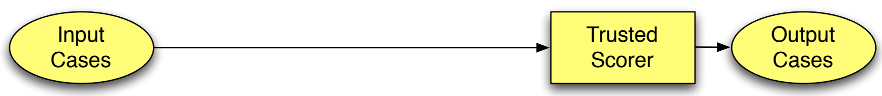

Fig. 1. General contingency analysis framework. The trusted scorer represents an algorithm that is trusted by the power grid community and is usually based upon simulating power flow.

Techniques for computing a numerical value for severity have been thoroughly researched in the power engineering community [4, 11]. Unfortunately, two factors complicate the matter: first, accurate simulation of a real electrical grid is not a trivial computation. While approximation techniques and iterative methods have been studied for over 40 years, precise solutions to the AC load flow equations still require on the order of seconds to compute on a typical server [8]. Second, the number of possible fault conditions which must be considered is staggering. Power engineers give the name "$N-1$ contingencies" to the set of possible grid conditions where exactly one major component has critically failed. Likewise, "$N-x$" refers to the set of all possible scenarios where x faults have occurred simultaneously. Current standards set by North American Electric Reliability Corporation (NERC), the primary U.S. regulatory body for electricity generation and transmission, requires utilities to ensure that the power grids they oversee can operate without failure under $N-1$ conditions, and that certain $N-x$ cases will not cause a system-wide blackout [12]. Given the limited number of cases considered, it is likely that current analysis will overlook many preventable catastrophic failures resulting from multiple outages. Unfortunately, one published model of only high-power transmission in the Western U.S. power grid contains almost 20,000 distinct transmission lines [16]. If one steps back to look at the numbers, $N-x$ contingency analysis for $x \leq 3$ would amount to well over 2.6 trillion possible failure conditions. Even on a modern parallel cluster of 1,000 servers, completely simulating every failure would take 8 years–the power engineering community is looking to get an answer in closer to a few *minutes.*

It should be clear that this is a severely resource-constrained problem. One way of overcoming this challenge is to throw out most of the contingency cases, running a full simulation on only the scenarios that are expected to be the most severe. We call the algorithm responsible a *filter.* Essentially, a filter is just a much simpler algorithm which computes a ranked list of all contingency cases by approximate severity. Then, based on the amount of computational resources available, the control center selects a small subset from the top of the list and runs a full simulation on each, producing an accurately scored final set. Historically, this has been a popular option in the power engineering community for two reasons: first, the number of catastrophic contingencies is usually very small, so removing faults which will have little overall impact can be effective at significantly reducing the input. Second, when a potentially-critical contingency is identified, an operator needs exact electrical information in order to take corrective action, so even if a filter was flawless at identifying severe cases, a full simulation would still need to be run.

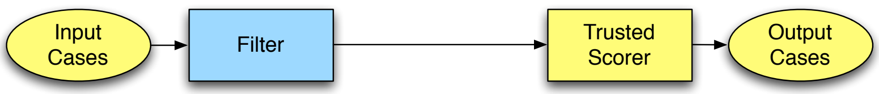

Fig. 2. Contingency analysis framework with filter. A filter reduces the number of input cases a trusted scorer must evaluate by predicting the severity of each case and only allowing the most probable through.

The simple framework in Figure 2 allows us to focus our efforts on a clean, abstract problem with well-defined elements. Over the next three sections we will introduce improvements to it that can subsequently be re-integrated into operational contingency analysis pipelines.

3 A Metric for Evaluating Filters

Influenced by the perspective of using this framework, one of the first observations we made was that, in spite of a long and well-documented history of research on filtering in contingency analysis [4, 1, 5, 15], there was a surprising dearth of work on metrics for quantitatively assessing the efficacy of a given filter (though some exceptions can be found [7]). Historically, research in this field was driven by the practical aspects of reducing the overall amount of computation, instead of a strict notion of accuracy.

To address this problem, we began by considering the behavior of an *ideal filter*. If a filter had perfect knowledge, then the set of output cases produced by a trusted scorer using that filter would be identical to the set of output cases produced by a trusted scorer using no filter at all. In other words, an ideal filter can predict with perfect accuracy the top k cases that a trusted scorer will produce and pass only those cases on. Using this definition, it is not difficult to invent a means for evaluating the efficacy of any given filter relative to the theoretical ideal. Essentially, the entire question boils down to choosing the best method for evaluating the similarity of two ranked lists–a question which has a large number of possible solutions, each with varying degrees of applicability to a given domain.

For the contingency analysis problem, we need only to compare the similarity of the top k elements in each list, since the assumption of our process framework is that there is only a limited amount of time to compute trusted scores. Moreover, we make a simplifying claim that a given contingency case is strictly more important than every other contingency case ranked below it in a given list (driven by the observation that in an operational contingency analysis setting, a human operator handles the output of a trusted scorer, and operators work in priority-order). Using these two assumptions, we note that by calling the top k cases selected using an ideal filter "True" and all other contingencies "False", we can directly apply any evaluation metric commonly used for binary classifiers. Thus, we can speak of the *precision* of a given filter by computing the number of contingency cases that appear in the top k of that filter and in the top k of the ideal filter.

It should be made clear that we do not claim that this adaptation of precision is globally applicable. On the contrary, we understand that the assumptions made here may not hold for other domains, and we embrace this diversity. Instead, the utility here is a means for taking an arbitrary comparison function over ordered sets and applying it to the filtering problem. Moreover, as we will soon see, this abstract definition of a filter metric enables further refinements.

4 Resource-Aware Filter Combination

In Section 2, we described the severe resource constraints on the problem of contingency analysis. The overwhelming majority of the computational cost of contingency analysis is spent in the simulation of the electrical behavior of the various scenarios under consideration. In terms of our abstract framework (Figure 2), since the trusted scorer is essentially derived from the electrical simulation we say that the cost of the entire pipeline can be represented by calculating the cost of the work done by the trusted scorer. Assuming that all other components have negligible cost might seem strong, but in practice, effective filters must have a completely different complexity in order to sufficiently reduce the dataset, and this assumption is quite reasonable.

Looking at the entire contingency analysis workflow, this observation produces an interesting effect: because a filter controls the size of the input set to a trusted scorer and the trusted scorer dictates the overall computational load, if given enough information, it is possible to construct a filter which takes as input the amount of resources available and produces an output set which will satisfy that constraint. We call such a filter *resource-aware*. It is worth reiterating that in our framework, such an algorithm need not be aware of its own computational cost but rather the induced effects of its output.

In general, a resource ceiling is usually given in time or compute cycle bounds. Unfortunately, a filter cannot directly control the amount of time spent by a trusted scorer on its workload; it can only adjust the number of cases it allows

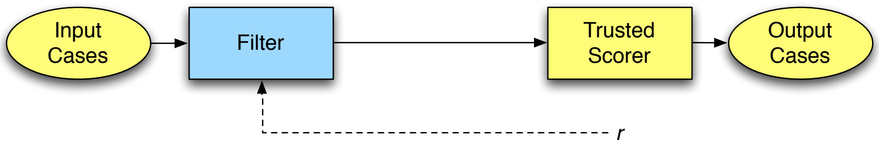

Fig. 3. Contingency analysis framework with resource-aware filter. A resource-aware filter uses information about the resources available to a trusted scorer (r) to adjust how many input cases are filtered out.

through. Thus, it is necessary to provide a cost function which takes a number of cases, k, and predicts the resources required to evaluate it, r. For our particular domain, we can make a reasonable assumption which trivializes this problem: the variance in the cost of computing any particular electrical grid scenario versus another is negligible–in practice, the cost of solving the load flow equations for a power grid is typically dominated by the size of the grid, not its state. Thus, our cost function is a simple linear relationship between the cardinality of the set of cases that a filter produces and the time a trusted scorer will take to evaluate them. In other words, we can assume $k \propto r$. For other domains, it is possible that a more complicated cost function may be necessary, but the overall principle remains the same.

With a clear definition of what constitutes a resource-aware filter in hand, we can extend the evaluation metric introduced in Section 3 to quantitatively evaluate these algorithms. Unfortunately, it quickly becomes evident, after experimenting with various candidates, that there exists inherent inconsistency between the efficacy of filters when measured across varying resource constraints. In other words, one method may be more effective than another when allowing 20% of the input dataset through, but the opposite may be true when allowing 60% through (see Figure 4). In practice, this means that one filter function rarely stands above the rest in all situations, but this need not be a stumbling block.

The natural solution is to use our evaluation metric as a predictive measure and adaptively choose the most appropriate filter from a set of candidates based on the current resource constraint conditions. Under these assumptions, we see a classic machine learning problem: we build a model of filter behavior with respect to available resources by evaluating the algorithm while varying the available resources. By creating such a performance model for each filter in our arsenal, we allow ourselves to dynamically select which algorithm to use while the contingency analysis pipeline is in operation. The resulting augmentations to our framework are shown in Figure 5.

We understand that we have left many aspects of model building for resource-aware filters untouched. While we will describe some of these as areas for future work later on, it is important to note that the contribution here is not specific results from machine learning on filters in contingency analysis (we have refrained from presenting them here), but rather the fact that by using our process frame-

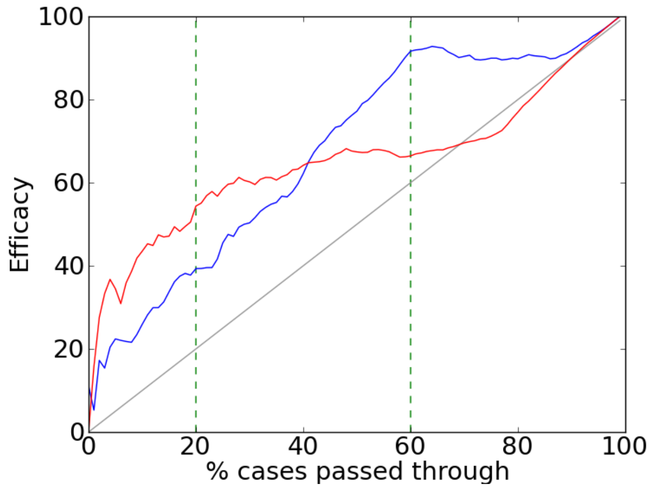

Fig. 4. Power grid example of efficacy crossover with respect to resource usage. To visually represent the concept of inconsistent filter performance, we plot the percentage of "bad" contingencies correctly captured (Y-axis) considering only a fixed number of cases (X-axis). Specifically, we use state information from a 765-bus, 982-line subset of the Western Electricity Coordinating Council (WECC) power grid (c.f. Jin et al.[9]) to assess efficacy of two filters: filter A is edge betweenness on a full, undirected graph representation of the dataset; filter B is edge betweenness on a partial, directed graph. Here we see A outperforming B at 20% but underperforming B at 60%.

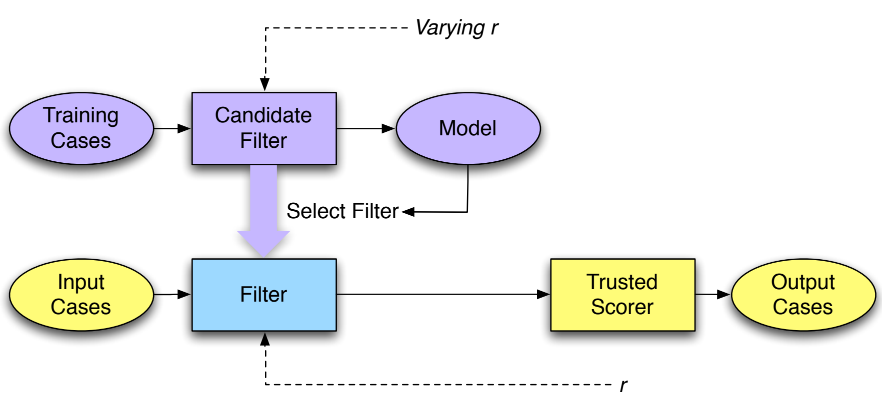

Fig. 5. Learning a resource model in the contingency analysis framework. By evaluating training data against a candidate filter, we can build a predictive model of efficacy vs. output size. This model can be used to select the most effective filter given the resources available to the trusted scorer, even in the presence of nonlinearity.

work we *enable* resource-aware machine learning. In other words, we claim these assumptions and abstractions are powerful in their own right, as they allow us to leverage a very large body of prior work in machine learning against this domain.

5 Multi-Criteria Optimization

In addition to massive computational demand, $N - x$ contingency analysis is complicated by the necessity of addressing various degrees of failure. We can observe that not only do various filters behave differently when operating on the $N-1$ problem versus the $N-2$ problem, but due to the substantial difference in computational complexity, some filters become completely infeasible to run (our assumption about neglecting the cost of filtering breaks down, inverting the cost model and rendering the filter unusable). Moreover, due to the lower probability of two simultaneous failures occurring and more lenient federal regulation, the relative priority of actually performing contingency analysis on $N - x$ changes drastically as x increases. In other words, it becomes crucial for operators to be able to allocate appropriate resources to different filters computing dissimilar problems based on their relative importance.

To simplify the discussion, we will start by generalizing the terminology somewhat. Instead of talking about concurrently solving the various contingency analysis problems, we say that there exists a number of *criteria* which must be evaluated using a shared, constrained amount of resources. For the power grid problem, information and code is often shared between different $N - x$ calculations. However, this overlap is primarily for convenience, not performance, and as such, we will assume criteria are effectively independent, and each must have its own separate pipeline (trusted scorer, filter, cost function, performance model, etc.). The only exception is that the computational resources consumed by one trusted scorer are shared with those of other criteria. We represent the layout of this problem visually in Figure 6.

Given the results from Section 4, we already have the tools available to solve this problem. For each filter, we build a predictive model which expresses the expected accuracy as a function of the level of resource reduction. Then we construct an objective function that takes as input a set of models (one for each criteria) and the available resources r and produces a set of resource allocations $\{r_A, r_B, ...\}$ that collectively optimize the operator's goal. When running the workflow operationally, each filter will then produce a set of cases to feed to its respective trusted scorer which should only consume the specified amount of resources. We augment our process framework with these new steps in Figure 7. We note that since the predictive models we generate are empirically-defined, we cannot make many assumptions about their behavior, and likewise, the same is true about an objective function which seeks to optimize some value using them. In other words, there is no efficient, general solution to creating the objective function we propose for all possible filters. However, part of the reason for this is that we have not placed any constraints on the goals which we seek to optimize;

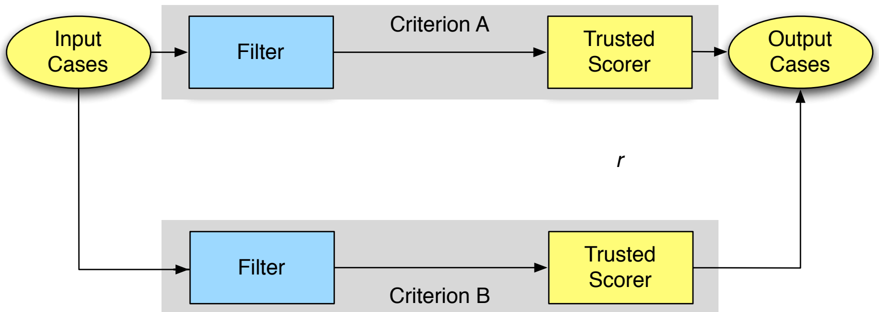

Fig. 6. Multi-criteria contingency analysis framework. Multiple independent criteria can be evaluated by constructing separate pipelines for each. The total available resources, r, will be shared when evaluating the trusted scorers.

with simplifying assumptions, it becomes relatively straightforward to produce a usable algorithm.

The most basic (and restrictive) assumption is to allow only a proportional allocation of resources. That is, the operator specifies a weight for each criteria indicating relative importance. A general objective function under this assumption is a simple linear weighting of the available resources–if filter A is given a weight of 0.25 and filter B is given a weight of 0.75, then one quarter of the available resources will always be given to the pipeline for criteria A and the remainder to pipeline B. Unfortunately, this assumption is blind; it does not take into account the relative efficacy of the filters being used. By using our predictive models, we can allow the operator to specify the importance of the *answers* they want, as opposed to the resource consumption. More precisely, if we let the function $M_A(r_A)$ be the efficacy of filter A as predicted by the model when producing an output set that will consume r_A resources, then we can formulate an expression for an objective function which prioritizes based on the expected behavior using a weight for each filter w_A.

$$w_A * M_A(r_A) = w_B * M_B(r_B)$$

$$r = r_A + r_B$$

Thus, our objective function becomes a solver for this set of equations. Keep in mind, however, that $M(r)$ is an empirically-derived model of the behavior of a filter, and as such, it cannot be assumed to be linear or even differentiable (and in practice, it is neither). As a result, analytical solvers are generally not useful for implementing objective functions. Using numerical methods for approximating a solution is generally sufficient, as the models $M(r)$ are predictive and therefore inherently entail an expectation of error, so a numerical solution need only be accurate to less than the margin of error of the predictive model (and our experience in the power grid domain has been that this is typically easy to achieve).

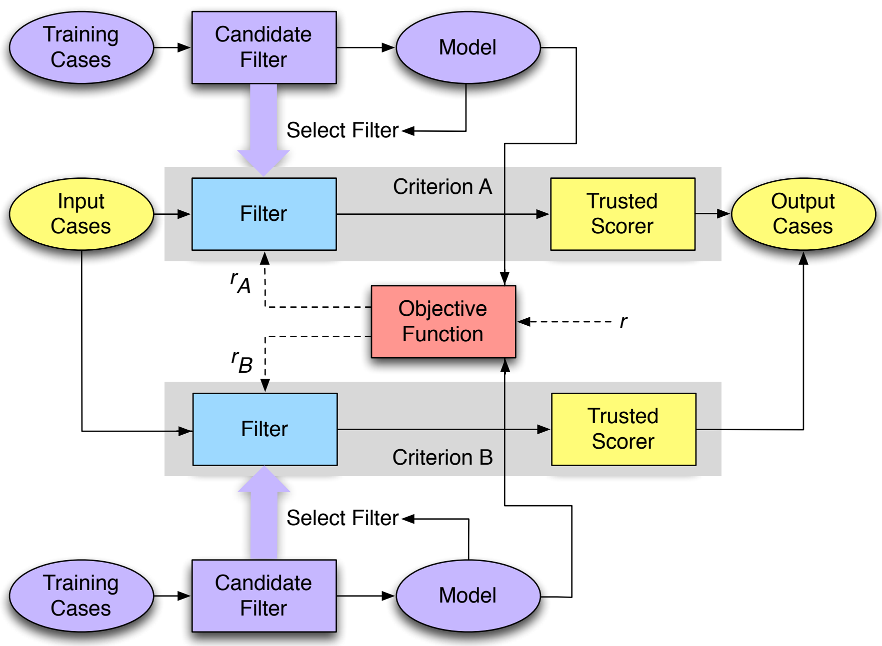

Fig. 7. Multi-criteria contingency analysis framework with resource-aware filters. By combining all the techniques introduced in this paper, we can intelligently apportion available resources to multiple pipelines by using predictive models to understand the efficacy/cost tradeoffs for each filter and an objective function to optimize the allocation based on an operator's priorities.

6 Related Work

There are a number of recent papers exploring topics in resource-aware mining; all of these discuss learning algorithms for building models where the algorithm is cognizant of resource availability and usage. The goal of these approaches is to use either minimal computation or to stay within available computational resources during the learning process and still produce a useful model. Two applications of this type of resource-aware algorithm include: building a model from within a sensor network, where battery charge may be a scarce resource [3, 13]; and stream mining where computational resources may be limited relative to the rate of streaming [14, 6]. We note that our approach differs significantly from these strategies in that our process aims to optimize the efficacy of an algorithm (in our case, a filter) based on the resource constraints of a separate, downstream procedure rather than algorithm itself. We consider the cost of training and assessing to be computationally negligible.

A recent paper introduced the idea of building decision-making modules in measurement systems that need to be deployed on resource-constrained platforms [10]. Their idea is to use genetic algorithms (GAs) and multiobjective optimization to guide the selection of support vector machine (SVM) models to deploy in a resource-constrained environment such as a wireless sensor network. Their method uses GAs to explore different parameter settings for an SVM that affects classification accuracy and computational complexity, and selects a finely-tuned SVM for deployment. Each SVM is targeted to a specific resource-constrained platform and is tuned to run with the available resources. In our case, the amount of available resources is not fixed at filter-tuning time; it is imposed after deployment and affects only accuracy, not the computational requirements. Recall that our assumption is that the cost of computing the filter is negligible relative to the cost of computing the trusted scorer. Our goal is to systematically assess the accuracy of a filter at differing availability of resources.

7 Future Work

In this paper we describe how we assess the quality of a filter that we simply placed into the framework. It might be significantly better to use one or more machine learning algorithm(s) to construct or tune a filter. An interesting study would be to compare a learned filter to one of the filters currently considered for this role such as betweenness centrality for $N-1$ contingency analysis [9] and group betweenness centrality for $N-x$, $x>1$ contingency analysis.

It may be advantageous to deploy multiple filters, some learned and some predefined, and devise a technique for combining the results of these filters into an overall rank ordering of contingency cases *a la* an ensemble technique. The intention would be to achieve greater overall performance by combining several algorithms than optimally selecting any single one. An interesting further study would be to compare several different ensemble techniques measured within our resource-constrained framework.

We have not explored the use of uncertainty quantification and how uncertainty values propagate through our framework. We understand (and have experienced) the error introduced by using a set of training cases to build the predictive models we use, but measuring this uncertainty could enable us to reason about the expected bounds of our optimizations. Understanding the uncertainty propagation might drive further improvement of filters and objective functions.

Unlike other fields of research, such as machine learning and data mining, standard datasets for power grids do not exist [2]. It would be valuable to explore and categorize different characteristics of power grid input and develop a repository of datasets that are representative of power grids in practice.

Finally, the contingency analysis problem can be viewed as a streaming problem with updates to the grid occurring in short time intervals, imposing a time constraint on decisions for corrective adjustments to the power grid in addition to the existing resource constraint. There is a reuse and prioritization aspect to a streaming model of the framework that may be worth exploring.

8 Conclusion

We have presented an abstract framework for intelligent filtering in the domain of power grid contingency analysis. Understanding the need for quantitative methods to systematically assess the quality of a filter algorithm, we introduced an evaluation strategy which adapts the notion of precision to fit our contingency analysis workflow. From there we observed that since the efficacy of a filter may be significantly different under varying conditions, we proposed a process for training and applying predictive models in order to select the best filter as a function of available computational resources. Finally, we extended our framework to support resource allocation using a multi-criteria optimization function across competing parallel workflows. We close by noting that while the practical application of our work to the contingency analysis problem has already been valuable, it is our hope that the generalization of the framework presented in this paper can offer sound tactics for using filtering as a solution for many resource-constrained environments as well as providing a basis for further opportunities to apply machine learning techniques to problems in the power engineering domain.

Acknowledgment

This work was funded by the Center for Adaptive Supercomputing Software - Multithreaded Architectures (CASS-MT) at the U.S. Department of Energy's Pacific Northwest National Laboratory. Pacific Northwest National Laboratory is operated by Battelle Memorial Institute under Contract DE-ACO6-76RL01830.

We would like to thank John Feo, ShuangShuang Jin and Daniel Chavarría-Miranda for their contribution to the software and lively discussions on the topic.

References

1. Albuyeh, F., Bose, A., Heath, B.: Reactive power considerations in automatic contingency selection. IEEE Transactions on Power Apparatus and Systems PAS-101(1), 107 –112 (Jan 1982)
2. Chassin, D.P., Posse, C.: Evaluating north american electric grid reliability using the barabsi-albert network model. Physica A: Statistical Mechanics and its Applications 355(2-4), 667 – 677 (2005)
3. Deshpande, A., Guestrin, C., Madden, S.: Resource-aware wireless sensor-actuator networks. Data Engineering 1001, 40 (2005)
4. Ejebe, G., Wollenberg, B.: Automatic contingency selection. IEEE Transactions on Power Apparatus and Systems PAS-98(1), 97 –109 (Jan 1979)
5. Ekwue, A.: A review of automatic contingency selection algorithms for online security analysis. In: Third International Conference on Power System Monitoring and Control. pp. 152–155. IET (1991)
6. Gaber, M., Yu, P.: A framework for resource-aware knowledge discovery in data streams: a holistic approach with its application to clustering. In: Proceedings of the 2006 ACM symposium on Applied computing. pp. 649–656. ACM (2006)
7. Halpin, T., Fischl, R., Fink, R.: Analysis of automatic contingency selection algorithms. IEEE Transactions on Power Apparatus and Systems PAS-103(5) (May 1984)
8. Huang, Z., Chen, Y., Nieplocha, J.: Massive contingency analysis with high performance computing. pp. 1 –8 (Jul 2009)
9. Jin, S., Huang, Z., Chen, Y., Chavarria-Miranda, D., Feo, J., Wong, P.C.: A novel application of parallel betweenness centrality to power grid contingency analysis. pp. 1 –7 (Apr 2010)
10. Marconato, A., Gubian, M., Boni, A., Caprile, B., Petri, D.: Accurate and resource-aware classification based on measurement data. IEEE Transactions on Instrumentation and Measurement 57(9), 2044–2051 (Sep 2008)
11. Mikolinnas, T., Wollenberg, B.: An advanced contingency selection algorithm. IEEE Transactions on Power Apparatus and Systems PAS-100(2), 608 –617 (Feb 1981)
12. North American Electric Reliability Corporation: Transmission operations standard top-004-2, `http://www.nerc.com/files/TOP-004-2.pdf`
13. Phung, N.D., Gaber, M., Rohm, U.: Resource-aware online data mining in wireless sensor networks. pp. 139 –146 (Mar 2007)
14. Shah, R., Krishnaswamy, S., Gaber, M.: Resource-aware very fast K-Means for ubiquitous data stream mining. Proceedings of 2nd International Workshop on Knowledge Discovery in Data Streams, ECML/PKDD (2005)
15. Singh, R., Srivastava, L.: Line flow contingency selection and ranking using cascade neural network. Neurocomputing 70(16-18), 2645–2650 (2007)
16. Western Electricity Coordinating Council: `http://www.wecc.biz`