Finding Needles in a Haystack: Line Event Detection on Smart Grid PMU Data Streams

Duc Nguyen, Scott Wallace, Xinghui Zhao

School of Engineering and Computer Science
Washington State University
Vancouver, WA USA
Email: {duc.nguyen, wallaces, x.zhao}@wsu.edu

***Abstract*—Smart Grid technology, in particular Phasor Measurement Units (PMUs) provide a mechanism for monitoring the state of a power system across a wide geographical area at high resolution and with high fidelity. These measurements form a large corpus of state information that power systems engineers and researchers can use to find and analyze interesting phenomena either post-hoc, or in real-time. In this paper, we present our work with machine learning to develop an event detector for use with the Bonneville Power Administration's (BPA's) current PMU installation. Our system can be used post-hoc or in real-time and focuses on identifying line faults in the data stream since these events can be easily verified by records BPA maintains. One challenge for machine learning algorithms is that the modern transmission systems are very often well-behaved. Since PMUs record measurements at 60 samples/sec and each PMU typically records up to 16 phasor signals, each PMU records upwards of 80 million measurements per day. Line events, in contrast, happen very rarely (on the order of 100 per month), at least on a transmission system such as BPA's. In this paper, we examine the performance of multiple classifiers within this power system domain. In addition to examining classifier performance on a set of validated line events, we perform a detailed analysis of false alarms and explore multiple methods for reducing false alarms in a real system.**

Keywords–Smart Grid; Phasor Measurement Unit (PMU); Machine Learning; Event Detection

I. INTRODUCTION

Smart-grid technologies offer the potential to increase efficiency and reliability of power transmission by facilitating precise monitoring and control over power demand and availability. Phasor measurement units (PMUs) are a key technology for monitoring the health and status of a power transmission system. These devices record system state variables (e.g., voltage and current phase angle and magnitude among others) at high data rates (typically 60 samples/sec). Critically, PMUs also provide precise timestamps on their measurements by using the Global Positioning System. This allows the variables measured across a wide geographical area to be synchronized.

Currently, PMUs are used primarily for monitoring purposes. However, in the future it is expected that they will play a much larger role in real-time operations and control. Some of these functions may include monitoring transmission lines for faults that may occur when lines contact one another or come into contact with a grounded object. Such occurrences typically result in an observable sag in voltage at nearby substations and may require relays to actuate to isolate the affected transmission line from the rest of the grid.

In this paper, we present recent work applying machine learning techniques to detect line faults. Our experiments are conducted on archival data from a large, active power transmission system spanning the Pacific Northwest region of the United States of America. The main contributions we present include: (1) a cascade classifier that improves upon both hand-built classification rules provided by a domain expert and upon a corresponding decision tree inferred by Weka's [1] J48 implementation of the learning algorithm C4.5 [2]; and (2) a detailed examination of the classifier's error rates and an evaluation of methods for reducing false positives (normal grid activity classified as a fault).

In the following sections, we begin with a brief review of related work using machine learning for fault detection in the grid. In Section III, we describe our initial efforts to detect line faults with machine learning. These approaches serve as a baseline for the contributions described in this paper. In Section IV, we describe our dataset and then follow in the subsequent section with a discussion of two new cascade classifiers and their relative performance. In Section VI, we focus our examination on the false positive rate and on ensuring that the tradeoff between true positives and false positive stays within reasonable limits. Finally, in Section VII, we examine the cascade on a significantly larger volume of data from normal operation (nearly 100 million examples) and compare the cascade's false positive rate to that of the baseline decision-tree classifier.

II. RELATED WORK

The increasing number of PMUs, as well as their high sample rates, present challenges for the traditional workflow of data processing [3]. As machine learning techniques have become the *de facto* standard for processing and analyzing large amount of data in other scientific fields, a variety of these techniques have also been applied to analyze PMU data for the purpose of recognizing synchrophasor signal patterns or signatures of events. In general, these approaches can be divided into two categories: unsupervised learning and supervised learning. The widely adopted techniques in these two categories are clustering and classification, respectively. The former is well suitable for identifying events with unknown signatures and the latter is particularly useful for classifying events based on a known taxonomy. Antoine et al. propose a clustering method based on PMU measurements to identify causes for inter-area oscillations [4]. In [5], the hierarchical clustering method is applied to identify dynamic event locations. The hierarchical clustering method has also been used to analyze disturbance events [6]. Clustering methods with unsupervised learning do not require labeled training data. However, once the clusters are generated, an expert's knowledge is often needed to interpret and compare the clusters [4].

In contrast to clustering, supervised learning approaches [7] can take advantage of labelled training data and identify known signatures or patterns without domain knowledge from experts. These approaches have also been used to detect interesting events from PMU data streams. For instance, Zhang et al. propose a classification method for finding fault locations based on pattern recognition [8]. The key idea is to distinguish a class from irrelevant data elements using linear discriminant analysis. The classification is carried out based on two types of features: nodal voltage, and negative sequence voltage. Similar classification techniques are used to detect voltage collapse [9] and disturbances [10] in power systems. Specifically, Diao et al. develop and train a decision tree using PMU data to assess voltage security [9]. Ray et al. built Support Vector Machines and decision tree classifiers based on a set of optimal features selected using a genetic algorithm [10]. Support Vector Machine-based classifiers can also be used to identify fault locations [11] and to predict post-fault transient stability [12].

Although significant effort has been focused on mining PMU data for detecting event signatures, very few projects use real PMU data collected from a wide area monitoring system. To the best of our knowledge, there is no previous work focusing on reducing false positives in classifying PMU data. Because of the stability of the power grid, line faults are rare phenomena. As a result, reducing false positives of the classification becomes critical. In the work presented in this paper, we examine real PMU data collected from the BPA's operational grid and propose several methods to reduce the number of false positives produced by supervised classification.

III. Baseline Fault Detection

In 2014, we described a simple set of decision rules for identifying line faults from their voltage sags [13]. These rules were developed using a theoretical foundation and validated on faults recorded over an 11 month period on the Bonneville Power Administration's transmission grid which covers a large geographic area in the Pacific Northwest of the United States of America.

Although the expert-built classification rules performed well (Table I, first row), they require a tremendous amount of effort to generate. Indeed, hundreds of pages of plots and data were generated for roughly 110 candidate faults. Clearly, the manual approach does not scale well, and as a result we have pursued machine learning approaches to leverage the data recorded by PMUs before and during these faults.

In our initial approach, we were able to show that the J48 decision tree learner was able to vastly improve upon the domain expert's hand crafted rules by: (1) leveraging a substantially larger amount of training data; and (2) improving upon the classic features directly measured by PMUs. We selected decision tree learning for these experiments because the learned representation is directly comparable to the expert's rules. Thus, we can identify how scaling up the data volume and changing the feature set contribute to performance independent of influences caused by changing the learned representation of the decision surface. While both the expert generated rules and the inferred decision trees perform classification well at PMUs closest to where the faults occur, the learned decision trees perform much better when used at an arbitrary PMU location (hence significantly higher recall values for J48 than the expert defined rules).

Table I illustrates these results. Boolean precision and recall refer to the boolean classification task of determining whether a particular moment in time should be considered a "line-event", or "normal operation". Accuracy is measured across all four possible classifications (three line-event classes: single line to ground fault (SLG); line-to-line fault (LL); and three-phase fault (3P); and the normal operating condition (N)). Macro precision and macro recall are macro-averaged values over the three line event types. We exclude the precision/recall figures from normal operating condition in the marco average since performance on normal data is best evaluated by the boolean performance characteristics. Table I illustrates a notable boost in recall for both the boolean and individual fault types cases (column two and five respectively). Moreover, while we can see that increasing the data (row 2) used for training the classifier improves recall, using a modified feature set also substantially improves the macro-recall of individual fault classes (column five).

TABLE I. Initial Performance of J48 vs Hand-Coded Rules

	Boolean Precision	Boolean Recall	Accuracy	Macro Precision	Macro Recall
Expert Rules	100	34.4	77.2	96.8	20.9
J48 Increased Data	97.2	97.8	96.5	78.2	53.0
J48 Increased Data + New Features	97.2	97.8	97.6	77.5	91.1

While the results do indeed show that both J48 classifiers make substantial improvements in recall, there is a tradeoff in precision. Most notably, boolean precision drops from 100% on the test-set for the expert defined rules to 97.2% for both learned classifiers. At one level, this seems like a reasonable tradeoff for the gains that are made on other metrics. However, a loss of almost 3% precision (column 1) means many false alarms; in a real operating environment, that value may be prohibitively high.

The remainder of this paper focuses on alternate classifiers and techniques that address the potential problem of false alarms in the J48 classifier listed above. We begin by discussing the evaluation dataset.

IV. PMU Dataset

For this study, we leveraged archival PMU data recorded by BPA between October 2012 and September 2013. We extracted examples of normal operation and each of the three fault types from within this sample. BPA verified 100 examples of line-events from within our sample using their operational database.

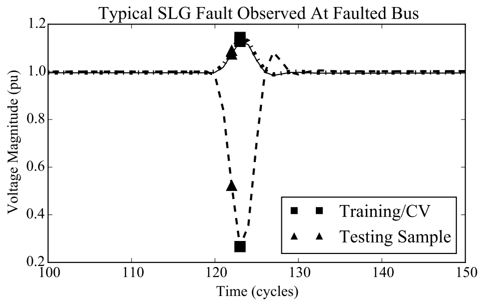

Figure 1. Typical voltage signature during a line event

Fifty-seven of these included log notes sufficient to determine a specific fault type (e.g., single-line-to-ground, line-to-line, or three-phase). The remaining faults were plotted and manually inspected to verify their voltage signatures and infer their type.

The voltage magnitude signature of a typical fault is illustrated in Figure 1. For any given fault, the PMUs reporting the largest voltage magnitude deviations from their steady state values tend to be physically close to the location of the fault. In contrast, PMUs located geographically distant from a fault may experience a minor deviation from steady state that is not easily discernible from normal variance. Unless otherwise indicated, the fault instances used to train and test the SVM classifiers described below use samples obtained across *all* PMUs with valid signals, regardless of the magnitude of the deviation experienced by each PMU. The procedure for obtaining these fault samples is as follows:

For each of the 100 fault events we found the moment in time where the measured voltage sagged to its lowest relative value. For most faults, PMUs report this moment as within 1 or 2 cycles (1/60 seconds) of each other. The most commonly reported time, for each fault and across all PMUs, was labeled the “moment of maximum voltage deviation” (square point in Figure 1).

Given the moment of maximum voltage deviation for each fault, we then collected examples using all PMU signals that measured three valid voltage phases at those moments in time. In total, the approach yields 7125 SLG examples, 474 LL examples, and 267 3P examples. To this set of 7688 fault instances, we added 11419 examples of normal operation randomly selected from across 800 minutes where BPA reported no line events occurred. Together, these samples comprised the training data for our Support Vector Machine (SVM) classifiers.

We created a test-set using the same 100 fault events but sampling them 1 cycle (1/60 second) earlier than in the training data (triangular point in Figure 1). To this, we added an additional 11419 samples of normal data obtained in the same manner as the normal data from the training set. Although the fault samples in our test set are not entirely independent of those in the training data, we believe that these points serve as a reasonable proxy for new archival data with unseen faults. Indeed, they may even be a more challenging test than new events since the selection methodology ensures that they will typically be smaller deviations from steady state than the points used in training.

Each example in the training and testing set consists of the three-phase voltage measurements for a single point in time, from a single PMU. Voltages are normalized by steady-state values calculated over a short window (10-30 cycles) that precedes each example. The voltage within the window is subjected to a smoothness check prior to computing the steady state; if this test fails, the window is moved backward in time relative to the sampling point until a suitably smooth voltage can be found as reference. In Figure 1, for example, when obtaining a measurement for the moment of maximum voltage deviation (square points at Time=122 cycles), the steady state window would initially be located so as to include cycle 121 since this point immediately precedes the measurement. However, cycle 121 deviates significantly from the mean value over the preceding 10 cycles. Thus, the smoothness criteria will not be satisfied until the window is backed up one cycle earlier so that it ends at cycle 120. This approach aims to compute a reasonable steady state value that is not adversely influenced by the voltage deviations experienced near a fault. We applied this technique to normalize all voltage measurements for both the training and testing data described above.

Using this dataset, we compare the historically best performing decision tree (row 3 in Table I) against classifiers built using SVMs in Section V. We then explore how our classifiers stack up when asked to classify a day of continuous data from 19 operational PMUs.

V. Cascades of SVMs

The main benefit of J48 is that its classification rules can be directly compared to expert rules and validated by power systems engineers. By itself, this is a substantial value. However, we anticipated that other learning algorithms may be able to improve upon J48’s performance. To this end, we examined using two-class SVMs [14] in three different configurations: a standard “one-against-one” approach for multi-class learning; configured in a 3-Stage SVM Cascade (illustrated in Figure 2); and configured in a 5-Stage SVM Cascade (illustrated in Figure 3).

The one-against-one method is the default approach for learning multi-class problems with SVMs in Python’s scikit-learn version 0.16 [15], which we leverage for our work. In our domain, there are four classes (three fault classes and the normal operation class). The one-against-one method learns six binary classifiers covering the $\binom{n}{2}$ class pairs. The classifiers then vote on the final prediction.

The 3-Stage and 5-Stage SVM Cascades, in contrast, learn individual binary SVMs representing a single decision node. Thus, the cascade acts similarly to a decision tree in that a series of SVMs which make progressive refinements to the set of potential classifications until a single classification is obtained. Unlike the one-against-one SVM which uses a single set of hyper-parameters for all binary SVMs that are learned, we allow each node in the Cascade to have independently set hyper-parameters which we obtained with a grid search over a subset of the training data.

The 3-Stage SVM Cascade was explored because it replicates the decision order performed by earlier J48 classifiers. The 5-Stage SVM Cascade was explored because we observed that making the fault/no-fault classification last, rather than first, may generate fewer false alarms. This is because different fault/no-fault decision boundaries could be found once the primary characteristics had a chance to group the examples into their most likely constituent fault type.

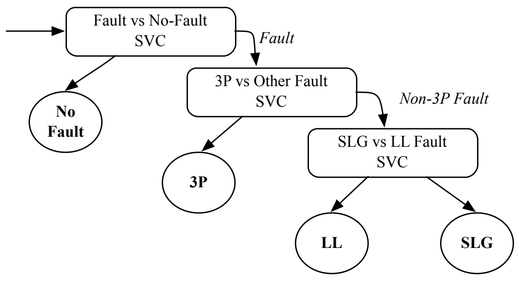

Figure 2. 3-Stage SVM Cascade

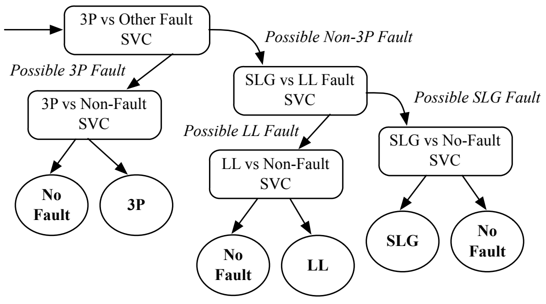

Figure 3. 5-Stage SVM Cascade

Confusion matrices for the original J48 classifier on the new test set, along with the confusion matrices for the three SVM variants are illustrated in Tables II and III respectively.

The J48 Classifier's performance, illustrated in Table II, shows high accuracy (95.5%), but a false positive rate of 4.9% (normal data classified as a fault: $(96 + 167 + 294)/11419$) that will be likely be unacceptable in a continuous operation deployment.

TABLE II. TESTING SET PERFORMANCE OF J48 CLASSIFICATION

		Predicted Class			
		SLG	LL	3P	NF
True Classification	SLG	6935	37	7	146
	LL	124	348	0	2
	3P	1	0	266	0
	NF	96	167	294	10862

The SVM classifier performance is illustrated in Table III. The One-vs-One configuration (left figure) and the 3-Stage SVM Cascade (middle figure) show relatively similar accuracy (94.3% and 92.9% respectively) as the original J48 classifier and likewise similar false positive rates (4.5% for both configurations). The 5-Stage SVM Cascade, in contrast, maintains an accuracy similar to the 3-Stage Classifier (93.0%), but reduces false positives to 1.7%, nearly a factor of 3 below all other classifiers examined.

VI. TOWARDS FULL-TIME OPERATION

The results from Section V illustrate that each of the various classifiers makes somewhat different tradeoffs in terms of accuracy and false positive/false negative rates. Before selecting one classifier for further study, however, we wanted to explore how the false positive rate was impacted by modifications to the training data.

Recall that a PMU measuring a nearby line event tends to produce a relatively large deviation from steady state voltage. In contrast, PMUs located relatively far from a fault location may measure a voltage deviation that is small enough to be nearly lost in the normal fluctuations of the system. Thus, one approach to reducing the false positive rate would be simply to ignore measurements from PMUs that are geographically distant from a fault location. Intuitively, this should not be problematic from a detection standpoint since our dataset contains a large number of PMUs over a wide geographical area (covering two states in the Pacific Northwest of the United States of America). Thus, removing a portion of the PMU measurements should still provide enough measurements to classify the event.

Figure 4 shows how each of the four classifier's false positive rate changes as a result of thresholding the fault instances. Specifically, the figure plots a representative point for each of the four classifier's performance as represented in the confusion matrices in Tables II and III. For the J48 classifier whose decision nodes are discussed in [13], we change the threshold on the first decision node which corresponds to the fault/no-fault distinction; we do not retrain the classifier as a whole. As the threshold is tightened, the net effect is to classify examples with small deviations (which tend to occur far from a fault location) as normal. We plot the impact of this modified threshold as a line (blue dashed line). Note that the true positive rate reported on the Y-axis is exactly that: the true positive rate on the *unmodified* test set. That is to say that although we are changing the threshold associated with the fault/no-fault classification in the decision tree, we are not changing the labels associated with the testing data, nor are we removing instances from the test set. The line on the plot shows that no modification can be made to the J48 threshold so as to obtain the same true positive/false positive rate as the 5-Stage Cascade.

To see if thresholding would permit either the One-vs-One SVM or the 3-Stage SVM Cascade to reach a similar performance profile as the 5-Stage SVM Cascade, we performed a similar analysis as for J48. Here, however, we applied the threshold to the fault instances in the training set, thereby removing the fraction of examples whose voltage deviations were more subtle than the threshold allowed. Again, this has the effect of removing measurements from PMUs geographically distant from the fault location from the training set. Once the new classifiers were trained, we then tested on the full testing set and plotted the resulting lines. The plot reveals that the One-vs-One SVM tends to outperform the 3-Stage SVM Cascade for false positive rates between roughly 0.8% and 4.5% respectively. However, like J48 and the 3-Stage classifier, it does not quite yield the same true-positive

TABLE III. CONFUSION MATRICES FOR: 1-VS-1 SVM (LEFT); 3-STAGE SVM CASCADE (MIDDLE); 5-STAGE SVM CASCADE (RIGHT)

		Predicted Class			
		SLG	LL	3P	NF
True Classification	SLG	7063	9	0	53
	LL	192	207	0	75
	3P	1	0	23	243
	NF	469	50	0	10900

		Predicted Class			
		SLG	LL	3P	NF
True Classification	SLG	6361	54	2	708
	LL	79	380	0	15
	3P	1	0	259	7
	NF	83	162	267	10907

		Predicted Class			
		SLG	LL	3P	NF
True Classification	SLG	6329	18	0	778
	LL	89	252	0	133
	3P	1	0	139	127
	NF	83	53	60	11223

performance as the 5-Stage classifier. The 5-Stage Cascade overall performs best showing the lowest false positive rate with the full training set (upward facing triangular point). For false positive rates between $0.008 - 0.017$ the 5-Stage SVM Cascade and the 3-Stage Cascade perform relatively similarly. However, the 5-Stage Cascade once again dominates performance for lower false positive rates.

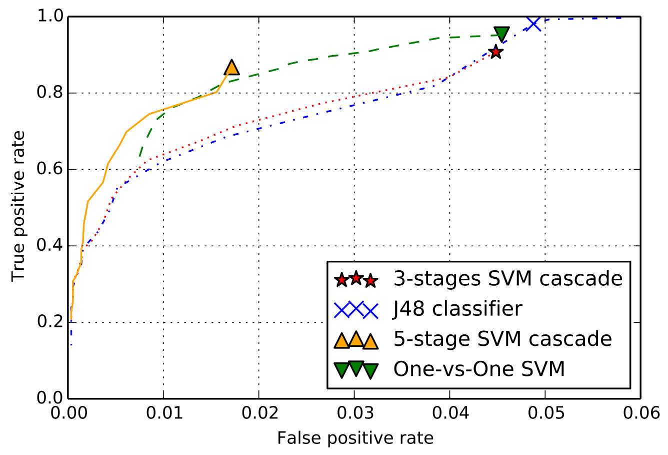

Figure 4. ROC curve for J48 and SVM classifiers

VII. LARGE TEST SET PERFORMANCE

The ROC curve in Figure 4 shows that no classifier outperforms the 5-stage SVM Cascade. In this Section we explore the 5-stage SVM Cascade's performance as we scale the test data to simulate a real operating environment. We expect that our previous testing is robust with respect to the system's ability to identify fault instances given that we have sampled an entire year's worth of line events that occur on transmission lines adjacent to one or more PMUs. However, our previous test set examined only a small sample of normal data. As a result, we expect that our estimate of false positive rate may be incorrect by a significant margin. Thus, in this experiment we focus on a contiguous 24 hours period in which no known line faults occur. Since line faults are relatively rare, this is not an atypical situation. Given this period of data, we then want to examine the performance of the 5-Stage SVM Cascade to determine how the false positive rate compares to our much smaller test set previously examined. To provide a baseline, we performed the same large scale test with the original J48 classifier.

We used signals from 19 PMUs across the grid over a contiguous 24 hour period. Each PMU reports 5,184,000 measurements over a 24 hour period, however one PMU is offline for 283 cycles (4.7 seconds) during the period we selected. Additionally, we use the first 10-cycles from each PMU to acquire a steady state value for each voltage measurement. In total this means that there are $(5.184 \cdot 10^6 \cdot 19) - 283 - (10 \cdot 19) \approx 98.5 \cdot 10^6$ individual examples to classify. However, from the analyst's standpoint, it is the moments of time that matter, not the individual measurements, so we consider only the unique moments where all PMUs report valid signals (i.e., $5.184 \cdot 10^6 - 293$ moments).

When run on these individual moments of time, the 5-Stage SVM Cascade classifier reports 15 false positive moments in time, but J48 reports an incredible 4,582,490 false positives. Our initial hypothesis was that one PMU may have been behaving erratically during the day we selected. To test this, we recorded the number of PMUs whose signals were classified as "fault condition" for each moment in time. We then applied a floating threshold so as to require multiple PMUs to corroborate each others' classification before marking a particular moment in time as a fault. We tested thresholds of 1,2,3, and > 3 PMUs. In the first case, we ask how many times exactly 1 PMU classifies a moment as a fault; in the last case, we ask how many times there are more than 3 PMUs classifying the same moment as a fault. These results are shown in Table IV. The first column shows how many moments cannot be classified (because they are used to initialize the steady state measurement or because a PMU is out of service); while the second column shows how many moments are classified as "normal" by all PMUs. The stratification demonstrates that J48 is not being adversely affected by a single PMU, since 75% of the day is classified as a fault by more than three PMUs.

Our next suspect was that J48 was adversely affected by the steady state measurement. Recall that one goal of our steady state computation was to ensure that we didn't inadvertently incorporate part of a fault into the window over which steady state value was calculated. As a result, our algorithm also ensured that moments in time labeled as a "line event" were not used in subsequent steady state calculations. This means that a contiguous string of false positives could result in the steady state measurement being significantly far back in time (and thus unlikely to be a precise measurement).

We re-ran the test using two variations of the steady state calculation, again recording the amount of corroboration between PMUs. The second steady state method used the same basic algorithm, but reported an error condition if the steady state window lagged more than three cycles behind the measurement point. This approach ensures that there was only minimal lag between the steady state window and the measurement, but may result in many errors (since there may be some periods that were not smooth enough to satisfy our steady state algorithm). Results from this approach are illustrated in Table V.

The third variation simply uses the previous 30 cycles for the steady state measurement, regardless of the smoothness of the measurements during that window, and regardless of the classification of data within that window. The attempt here was to ensure that the steady state window is always close in time to the measurement point and to reduce the time in which the steady state is incalculable. Since the window would always keep pace with the measurement, we expected that transient noise may create some false positives, but eventually the signal would smooth out and classification would continue regularly. We increased the window length from 10 to 30 cycles to avoid being overly influenced by isolated measurement outliers. Results from this approach are illustrated in Table VI.

TABLE IV. FALSE POSITIVES FOUND BY CORROBORATED CLASSIFICATIONS

	Out of Service	0 PMUs Report Fault	1 PMU Reports Fault	2 PMUs Report Fault	3 PMUs Report Fault	> 3 PMUs Report Fault
5 Stage SVM Cascade	293	5,183,692	10	3	0	2
	0.0057%	99.9941%	0.0002%	0.0001%	0.0000%	0.0000%
J48	293	601,217	139,196	226,854	285,764	3,930,676
	0.0057%	11.5976%	2.6851%	4.3760%	5.5124%	75.8232%

TABLE V. METHOD 2: LAGGED STEADY STATE WINDOW PRODUCES AN ERROR

	Out of Service	0 PMUs Report Fault	1 PMU Reports Fault	2 PMUs Report Fault	3 PMUs Report Fault	> 3 PMUs Report Fault
5 Stage SVM Cascade	333	5,183,653	10	2	0	2
	0.0064%	99.9933%	0.0002%	0.0000%	0.0000%	0.0000%
J48	1,763	5,181,655	527	26	11	18
	0.0340%	99.9548%	0.0102%	0.0005%	0.0002%	0.0003%

TABLE VI. METHOD 3: 30-CYCLE SS WINDOW KEEPS PACE WITH MEASUREMENT

	Out of Service	0 PMUs Report Fault	1 PMU Reports Fault	2 PMUs Report Fault	3 PMUs Report Fault	> 3 PMUs Report Fault
5 Stage SVM Cascade	313	5,183,669	13	3	0	2
	0.0060%	99.9936%	0.0003%	0.0001%	0.0000%	0.0000%
J48	313	5,181,968	1,440	130	41	108
	0.0060%	99.9608%	0.0278%	0.0025%	0.0008%	0.0021%

The results across all three steady state variations indicate that the 5-Stage SVM Cascade is relatively insensitive to the method of steady state calculation, while J48 performance is much more sensitive, even as the amount of corroboration required between PMUs is increased. Overall, the results from the 5-Stage Cascade are quite promising in this streaming environment. Across the 24 hour period, there are only 2 moments in time (regardless of the steady state method employed) that are incorrectly classified as faults by more than three PMUs. This false positive rate is 0.0000386% — more than four orders of magnitude less than expected given the results from the original test set.

VIII. CONCLUSIONS

In this paper, we have demonstrated a 5-Stage Cascade of SVMs within the PMU line fault domain. Our cascade offers a better performance profile than previously reported J48 classifiers for the same domain. Critically, we also have shown that the 5-Stage classifier achieves very low false positive rate on a 24 hour period of data containing almost 100 million examples. For future work, we are exploring the potentials of using unsupervised learning to characterize line events and unknown events on the smart grid.

ACKNOWLEDGMENTS

This work is partially supported by the DoE / the Bonneville Power Administration through the Technology Innovation Program (TIP #319). The authors would also like to thank BPA for the use of archival PMU data.

REFERENCES

[1] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD explorations newsletter, vol. 11, no. 1, 2009, pp. 10–18.

[2] R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[3] Americans for a Clean Energy Grid, "Synchrophasors," http://cleanenergytransmission.org/wp-content/uploads/2014/08/Synchrophasors.pdf, 2014, [Retrieved: September, 2015].

[4] O. Antoine and J. C. Maun, "Inter-area oscillations: Identifying causes of poor damping using phasor measurement units," in Power and Energy Society General Meeting, 2012 IEEE. IEEE, 2012, pp. 1–6.

[5] K. Mei, S. M. Rovnyak, and C.-M. Ong, "Clustering-based dynamic event location using wide-area phasor measurements," IEEE Transactions on Power Systems, vol. 23, no. 2, 2008, pp. 673–679.

[6] O. P. Dahal, S. M. Brahma, and H. Cao, "Comprehensive clustering of disturbance events recorded by phasor measurement units," IEEE Transactions on Power Delivery, vol. 29, no. 3, 2014, pp. 1390–1397.

[7] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in Proceedings of the 23rd International Conference on Machine Learning. ACM, 2006, pp. 161–168.

[8] Y.-G. Zhang, Z.-P. Wang, J.-F. Zhang, and J. Ma, "Fault localization in electrical power systems: A pattern recognition approach," International Journal of Electrical Power & Energy Systems, vol. 33, no. 3, 2011, pp. 791–798.

[9] R. Diao, K. Sun, V. Vittal, R. O'Keefe, M. Richardson et al., "Decision tree-based online voltage security assessment using pmu measurements," IEEE Transactions on Power Systems, vol. 24, no. 2, May 2009, pp. 832–839.

[10] P. K. Ray, S. R. Mohanty, N. Kishor, and J. P. Catalão, "Optimal feature and decision tree-based classification of power quality disturbances in distributed generation systems," IEEE Transactions on Sustainable Energy, vol. 5, no. 1, 2014, pp. 200–208.

[11] R. Salat and S. Osowski, "Accurate fault location in the power transmission line using support vector machine approach," IEEE Transactions on Power Systems, vol. 19, no. 2, May 2004, pp. 979–986.

[12] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," IEEE Transactions on Power Systems, vol. 26, no. 3, 2011, pp. 1474–1483.

[13] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao, and X. Liang, "Smart grid line event classification using supervised learning over PMU data streams," in Proceedings of the 6th International Green and Sustainable Computing Conference (IGSC 2015), Dec 2015, pp. 1–8.

[14] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, 1995, pp. 273–297.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, 2011, pp. 2825–2830.