A Hybrid Machine Learning Framework for Enhancing PMU-based Event Identification with Limited Labels

Haoran Li, Yang Weng  
Department of Electrical, Computer and Energy Engineering  
Arizona State University, Tempe, AZ, USA  
Email: {lhaoran,yang.weng}@asu.edu

Evangelos Farantatos, Mahendra Patel  
Electric Power Research Institute  
Palo Alto, CA, USA  
Email: {efarantatos,mpatel}@epri.com

Abstract—The energy industry is experiencing rapid and dramatic changes on both the generator side and the load side, necessitating faster, more accurate, and robust event detection methods for situational awareness. Growing installations of PMU devices that provide high resolution synchronized measurements combined with the advancement of artificial intelligence and big data analytics techniques have recently attracted the R&D community interest. Some supervised learning techniques have been proposed using PMU measurements, however, they are facing challenges in 1) limited interpretability, 2) biased learning models/results, and 3) insufficient labeled data for learning. To address these issues, we propose a machine learning-based framework for physically-meaningful interpretability, hybrid-learning method with indexes, and a flexible data-preparation approach. Specifically, a thoroughly designed feature selection method is proposed for discovering event signatures. Then, a hybrid machine learning process is constructed to reduce biases of different machine learners due to their diversified working mechanisms. Finally, we propose to utilize unlabeled data via semi-supervised learning and add strategic event data via active learning, e.g., simulations. The goal is to significantly improve the supervised learning results via computational efficient techniques. Extensive simulations are conducted using a commercial power system dynamics simulator and synthetic realistic transmission grid models. Significant improvements are observed via hybrid supervised learning methods, semi-supervised learning, and active learning.

Keywords—Event identification, phasor measurement unit, feature selection, hybrid learning, semi-supervised learning, active learning.

I. INTRODUCTION

The power system is undergoing a dramatic transformation in both function and form. For example, the ongoing deployment of renewable energy technologies and electric vehicle techniques have the potential to offer cleaner energy [1], [2]. But, successful and reliable integration of new components for reliable transmission grid operation poses fundamental challenges. Without appropriate monitoring and control, even a small-scale uncertainty could destabilize the local grid and cause reliability issues. Such issues can lead to power outages or blackouts, which may cause a loss of millions to billions of dollars. Therefore, a highly active and accurate fault diagnosis process is necessary to accommodate increasing uncertainties in the transmission grid.

Meanwhile, the power system has a consistently increasing number of Phasor Measurement Units (PMUs) and other devices with high-resolution and synchronized measurements, providing opportunities to synchrophasor-based applications both for offline and online environment [4]–[7]. An area of research that has recently attracted the interest of the industry is the application of data mining and machine learning techniques for providing situational awareness to the system operators [8]. One example is data-driven event identification, where pattern recognition techniques could classify normal (typical) from abnormal (atypical) operating conditions, and provide guidance to system operators and engineers to further investigate whether an abnormal operating condition could result in a security threat to the system or not.

PMU-based data-driven methods for event identification that have been proposed in the literature can be categorized into two types [4], namely the signal-analysis approach and the machine learning approach. Signal-analysis approach views PMU data as time-series signals and applies signal processing techniques, such as wavelet transformation [9], Swing Door Trending (SDT) [4], and window-based threshold [10]. These methods focus on detecting the abnormal segment of signals but lack the ability to accurately identify the types and locations of events.

For machine learning methods, many studies focus on extracting features, such as the statistics and underlying constraints among different variables. Specifically, these studies target at features with large variances for indicating event type and location. For example, [11] utilizes Principal Component Analysis (PCA) to find the PMU measurements with large variance. [12] builds the event-type dictionary via principal components, which is used subsequently for event localization. However, such a variance-based feature selection approach is insufficient to identify events accurately, due to various system setup factors, e.g., unsymmetrical loopy structure, unevenly located voltage regulators, different system control signals, and intermittent renewable generations. In general, this method is not totally meaningful in the physical system, and the limited physical interpretability makes their methods non-robust to be applied in various grids.

There are also machine learning methods that directly conduct supervised learning by assuming the availability of a large number of event labels. Their mechanism is to find a mapping between PMU measurements and labels, where labels can be event types and locations. For example, there are works
on utilizing Support Vector Machine (SVM) [8], Decision Tree (DT) [13], Artificial Neural Network (ANN) [14], [15], and Long Short-Term Memory (LSTM) units [16]. However, these supervised learning methods may have their own preferences based on their learning principles. For example, they may return different solutions for the same type of training data and testing data. Therefore, we need to obtain a robust learning algorithm with indices to avoid biased learner and quantify the uncertainties in the machine learning algorithm. Further, neural network-based learning methods [14]–[16] lacks physical interpretability due to its large amounts of parameters and highly non-linear nature. For example, [16] inputs all the current, voltage and active power to the LSTM with a complex structure but can’t explain why we should construct a structure like this and whether this structure is still applicable in other networks or not.

Additionally, the supervised learning algorithms lack the ability to work well with limited training data set. Due to the fact that recorded events may sparsely occur in the historical PMU data stream, the majority of events is unrecorded or unknown. For example, [17] claims 1013 events based on a utility’s PMUs from 2001 to 2010, but only finds 84 events in the utility log files. Further, it is hard to find exactly the same scenarios for supervised learning, due to power system nature on varying load conditions. Unrepresentative training data set along with limited labels indicate a high risk of misclassification in the event detection.

In summary, the recently proposed learning methods are facing challenges in 1) limited interpretability, 2) biased learning models/results, and 3) limited labeled data for good performance. To address these issues, we propose a machine learning-based framework for a physically meaningful identification process, hybrid-learning methods with indexes, and a flexible data-preparation database.

Specifically, we propose to discover the event signature via a carefully designed feature selection metric. In such a metric, we embed both the fault type information, e.g., clustering number in different event types, and location information, e.g., network location. After feature selection, we input training data into different machine learning methods, such as decision tree, and K nearest neighbors, naive Bayes, logistic regression, support vector machine. After learning the models, they are applied to input data provided by PMUs. An index is then generated via entropy to quantify the uncertainty associated with the voted results on detection, localization, and event type. Such a process can efficiently balance the biases of different machine learners due to their diversified working mechanisms.

Finally, we have focused on improving the data for learning, so that the machine learning method can handle cases with limited labeled data [17]. For this purpose, we propose the semi-supervised learning to utilize unlabeled data and add strategical event data via active learning, e.g., simulations [18]. The goal is to significantly improve the supervised learning results via relatively cheap resources.

Thoroughly designed and extensive experiments are conducted on General Electric’s Positive Sequence Load Flow (PSLF) synthetic realistic power system models [19], [20]. The proposed method is compared with a benchmark against several classical machine learning methods. Significant improvements are observed in our hybrid supervised learning method, semi-supervised learning approach, and active learning method. Interpretation visualizations are displayed simultaneously for the confidence of system operators.

The rest of the paper is organized as follows. The problem formulation is in Section II. The proposed supervised learning approach is shown in Section III. IV shows how to enhance supervised learning with more data via semi-supervised learning and active learning. Section V shows the numerical results. Section VI concludes the paper.

II. Machine Learning Modeling

In order to formulate the learning methods for event identification, we need to describe the time series data set provided by PMUs and the associated label data set. For example, if there is an event in the historical data, we will store the event data in a matrix $X \in \mathbb{R}^{A \times B}$, representing the PMU measurements in $A$ time slots with a number of $B$ different measurement types. $\mathbb{R}$ represents the set of real numbers. If this event happens at time $i \in \{1, \ldots, C\}$ and event type $j \in \{1, \ldots, D\}$, we create a label $y_{i,j} = 1$ in the label matrix $Y$. If there is no event for type $j$ at time $i$, $y_{i,j} = 0$. Here, we extend the definition of the event type to include information on event location for avoiding over-complicated notations. For example, the same type of event happening at different locations will have different $j$’s in our setup. To associate $X$ with $y_{i,j}$, we rename $X$ as $X^{(i,j)}$.

1) A supervised learning problem:
- Problem: Event Identification via Supervised Learning
- Given: PMU data of $X^{(i,j)}$ and the label matrix $Y$.
- Find: The mapping rule of $f : X^{(i,j)} \rightarrow y_{i,j}$.

If there are unlabeled data going beyond the time period $i \in \{1, \ldots, C\}$ of the labeled data, we can define the time period of unlabeled data to be $i' \in \{C+1, \ldots, C''\}$. This creates unlabeled input data of $X^{(i',j)}$. Therefore, we have the following definition for utilizing unlabeled data.

2) A semi-supervised learning problem:
- Problem: Event Identification via Semi-Supervised Learning
- Given: Labeled PMU data of $X^{(i,j)}$, the associated label matrix $Y$, and the unlabeled PMU data of $X^{(i',j)}$.
- Find: The mapping rule of $f : X^{(i,j)} \rightarrow y_{i,j}$.

Finally, we can use $X_L$ to represent the labeled data and $X_U$ to represent the unlabeled data in general.

III. Supervised Learning

In this section, we focus on designing a physical meaningful feature selection method and a hybrid machine learning method with confidence indices.
A. Feature Selection via Physically Meaningful Metric Design

Feature selection for PMU-based supervised learning is important for the following reasons: firstly, due to “insufficient event data” and “insufficient labeling”, we can’t treat all the PMU measurements as features. Specifically, the small number of data can’t make the high-dimensional model with a large number of parameters fully trained, which prevents us to find the optimal values in the functional space. Secondly, the matrix form of PMU data possesses a low-rank property [11]. Therefore, feature selection is a must to filter useless information and capture the patterns of different events. Thirdly, a feature selection process is also beneficial for shortening the training and testing time.

The key question in feature extraction lies in how to extract the most meaningful features while significantly reducing data dimensionality. For event detection, our target for feature selection is to choose those who are mostly affected during the event time. It is proposed that features are selected based on the percentage change in the measurements. For example, if an event happens at time $a$, the percentage change between time slot $a-1$ and time slot $a$ will increase, leading to the following criterion

$$\frac{|x_{a,b} - x_{a-1,b}|}{x_{a-1,b}} > \epsilon_1, \forall 1 \leq b \leq B. \quad (1)$$

where $\epsilon_1$ is a constant threshold. Such a method is called the filter approach, looking into the input data only for feature selection and ignoring possible inductions. Then, we can obtain $X^{(i,j)}$ to represent data selected via (1). One can improve the filter-based approach from above by principal component analysis (PCA), e.g., via singular value decomposition (SVD).

$$X^{(i,j)}_S = U\Sigma V^T, \quad (2)$$

where $V^{[1:k]}$ represents the first $k$ column vectors of matrix $V$. $F^{(i,j)}$ is the selected features and they can be vectorized as a training sample $f^{(i,j)}$.

Due to the manually-chosen values of $\epsilon_1$ and $k$, such a feature selection process may not be totally useful for the supervised learning method later on. Also, the filter method does not take into account the physical network structure for localization.

Therefore, instead of the filter approach, we investigate the wrapper approach, which evaluates the selection of feature subset according to a certain induction algorithm. To use such an approach and embed physical laws for meaningful interpretation, we define the following metric based on regularization to incorporate physical network distance for localization. Let $e^{(i,j)}$ denote the $j^{th}$ type of event for the $i^{th}$ time, we have

$$d(e^{(i,j)}, e^{(k,l)}) = \|f^{(i,j)} - f^{(k,l)}\|_2 + \lambda||ind^{(j)} - ind^{(l)}||_2, \quad (3)$$

where $\lambda$ represents the penalty term and $ind^{(j)}$ represents the selected index vector for the $j^{th}$ event: if the feature is selected, the corresponding index in $ind^{(j)}$ is 1 and the rest are 0s. This metric can distinguish different events as well as keep the same event under different loading conditions with close values.

With the metric definition in (3), we conduct the following supervised learning algorithm in the wrapper approach. Firstly, we find the optimal centroid $z^*_j$ for the $j^{th}$ event:

$$z^*_j = \arg\min_{z_j} \sum_{i=1}^{c_j} d(e^{(i,j)}, z_j), \quad (4)$$

where $c_j$ is the total number of $j^{th}$ event. Since the metric in (3) is the summation of two Euclidean distances, the optimal (4) is the average among all the events of the $j^{th}$ type.

Then, we can form the metric-based supervised learning for feature extraction. Specifically, for a new event $e^{(1,h)}$ in the testing dataset, we have

$$h^* = \arg\min_{h \in \{1,2,\ldots,D\}} d(e^{(1,h)}, z^*_j), \quad (5)$$

where $D$ is the total number of event types. (5) gives us a predicted label for event $e^{(1,h)}$. Comparing all testing data’s predicted labels to true labels, we can obtain a testing accuracy for this supervised learning.

Finally, the metric based wrapper method is implemented as follows. First, we use different pre-defined threshold $\epsilon_1$ and $k$ in (1) and (2) to obtain different centroids of events, i.e., different classifiers. Subsequently, (3), (4), and (5) help to identify each event on a testing data set. Comparing the accuracy of the testing data set among different classifiers and we will acquire optimal hyper-parameters $\epsilon_1^*$ and $k^*$. Finally, they are used to select informative $k^*$ features that are utilized in the following supervised learning approaches.

B. Hybrid Supervised Learning

After feature selection, supervised learning can be used to study the relationship between PMU measurements and related events. In this work, we consider using different learning methods as candidates for making event decisions based on incoming PMU data. For example, Fig. 1 shows the process of the proposed hybrid machine learning methods. From the
left side, the hybrid learning method imports event log file, including historical PMU measurements and labels, into different machine learning models. Each machine learning model trains its classifier separately. To avoid biases from different learning models, the process in Fig. 1 not only combines the result of different models for event identification, but also provides an entropy-based index to measure the confidence of our voted label. The index $E$ is shown below:

$$E = 1 + \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} p(n,k) \log p(n,k) / \log(M),$$  \tag{6}$$

where $M$ is the number of classifiers, $N$ is the total number of testing samples, $K$ is the number of voting types and $p(n,k)$ is the percentage of votes for label $k$ in the $n^{th}$ testing samples. For each testing example, if different machine learning methods vote the same, we will obtain an entropy of 0. So, the index is 1, giving 100% confidence. If each classifier vote for a different label, we will obtain an entropy of $\log(M)$. With normalization and subtraction in (6), the index is 0 or 0%, showing that we do not have confidence in this estimate.

While the framework can incorporate any supervised learning method for event detection, localization, and type differentiation, we conduct in this work the following machine learning methods, which are representatives in machine learning philosophy. For example, their advantages and disadvantages are summarized in Table I.

1) Decision Tree: Decision Tree (DT) has a tree structure, whose nodes represent features, making the decision process easy to be interpreted. Each node in the tree helps the decision process, when the new data comes. To train a generalized tree from the training data set, a normal way to decide which feature to be associated with the next node is to maximize the information gain (IG):

$$x^* = \arg \max \ IG(D, X),$$  \tag{7}$$

where we assume that feature $x$ has $K$ values. If the current sample set is $D$ and we use $x$ as the next node, the information gain (IG) is:

$$IG(D, x) = H(D) - \sum_{k=1}^{K} p(x_k)H(D^k),$$

$$H(D) = -\sum_{k=1}^{|Y|} p_k \log(p_k),$$

$$p(x_k) = \frac{|D_k|}{|D|},$$

where $x_k$ is the $k^{th}$ value of feature $x$ and $D^k$ are the samples that have this feature value. $p_k$ is the percentage of each label $k$ in the data set $D$. The operator $|\cdot|$ is counting the number of elements of the objective and $H(D)$ represents the entropy of data set $D$.

2) $K$ Nearest Neighbors: $K$ Nearest Neighbors (KNN) is an instance-based machine learning methods. For a testing data set, the KNN calculates the distance between the testing sample and all the training samples. Then, the $K$ nearest neighbors are counted and vote for the label of testing data. For example, we can use Euclidean distance to calculate sample distance for KNN:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{K} (x_i(k) - x_j(k))^2}. \tag{8}$$

3) Naive Bayes: Such method tries to maximize posterior $P(Y|X) \sim P(Y)P(X|Y).$ \tag{9}$$

Following the conditional independence assumption, we have

$$P(X|Y) = P(X_1, X_2, \cdots, X_n|Y) = P(X_1|Y)P(X_2|Y) \cdots P(X_n|Y). \tag{10}$$

We can obtain the parameters $P(Y = y_k)$. $(1 \leq k \leq K, K$ is the total number of labels) and $P(X_i = x_i|Y = y_k)$ ($x_i$ is a value of random variable $X_i$) via maximum likelihood estimation (MLE) on the data set. When a new sample $\{x_1, x_2, \cdots, x_n\}^T$ comes, we can predict its label:

$$\hat{y} = \arg\max_{k \in \{1, 2, \cdots, K\}} P(Y = y_k) \prod_{i=1}^{n} P(X_i = x_i|Y = y_k). \tag{12}$$

4) Logistic Regression (LR): LR tries to maximize the conditional data likelihood

$$w^* = \arg\max_w \sum_{k=1}^{K} \ln P(Y^k|X^k, w), \tag{13}$$

where $K$ is the total number of training samples. The formula of the conditional probability has a sigmoid form

$$P(Y = 0|X) = \frac{1}{1 + \exp(w_0 + \sum w_i X_i)}. \tag{14}$$

5) Support Vector Machine (SVM): SVM tries to find the decision boundary

$$w^T x + b \tag{15}$$

with the maximum margin

$$w^* = \arg\max ||w||. \tag{16}$$

Since we assume data is distributed on the two outer sides of margins, we have the following constraints for the optimization problem in (15).

$$y_i(w^T x - b) \geq 1. \tag{17}$$
Different event clusters in the
Fig. 2. Testing the cluster assumption of semi-supervised learning in event identification problem.

If both the labeled and unlabeled data are available, we can

This assumption is valid in power system event identification. If we assume that all available PMU measurements form a high-dimensional feature space, different events, e.g., event type and location, will create different impacts on the system. Therefore, these diversified events create different clusters of high-dimensional points in the feature space. To illustrate this idea, Fig. 2a shows a network, where we trip different lines in different colors and obtain different events and associated PMU measurements. After conducting PCA to visualize the consequence in feature space, we obtain a 3-D visualization in Fig. 2b. It shows that different events create different clusters clearly, supporting the clustering assumption of semi-supervised learning in power system event identification. Finally, we also conduct different events together. Separable clusters are observed similarly to Fig. 2b.

To utilize both labeled and unlabeled dataset, there are many semi-supervised learning methods. Two popular methods are self-training and co-training approaches, where self-training makes the classifier use its own predictions to train itself and co-training employs two classifiers to train each other with the most confident prediction labels they feel. In this paper, we choose self-training since event identification does not need two classifiers in co-training. A popular model for such learning is the generative probabilistic model, where a Gaussian mixture model is calculated between the observation and the label. Then, the objective function attempts to learn the parametric vector.

Specifically, the class distribution $P(x|y)$ forms a model family $\{P(x|y, \theta)\}$ and the class priors are denoted as $P(y|\pi) = \pi_y$. For any fixed parameter estimator $\hat{\theta}$, $\hat{\pi}$, we can estimate $P(y|x)$ via

$$P(y|x, \hat{\theta}, \hat{\pi}) = \frac{\hat{\pi}_y P(x|y, \hat{\theta})}{\sum_{k=1}^K \hat{\pi}_{y_k} P(x|y_k, \hat{\theta})}, \tag{18}$$

where $K$ is the number of labels.

If both the labeled and unlabeled data are available, we can
obtain the parameters $\hat{\theta}$ by maximizing the joint log likelihood

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{C} \log(\pi_{y_i} P(x_i|y_i, \theta)) + \sum_{i=1}^{C'-C} \log \sum_{k=1}^{K} \pi_{y_k} P(x_i|y_k, \theta), \quad (19)$$

where $C$ is the number of labeled samples and $C - C'$ is the number of unlabeled samples. (19) can be solved via Expectation Maximization (EM) algorithm or direct gradient descent [22].

### B. Active Learning via Simulation Enhancement

During our study, we observe that not all the unlabeled events can give a positive impact on the final training process. For example, some events may be a small system disturbance that cannot be classified into any existing labels in the log file. Meanwhile, some unlabeled events, if misclassified in the iteration of semi-supervised learning, may lead to a decrease of classification accuracy.

Active learning works by requesting a small number of true class labels for unlabeled events it selects for the biggest impact [23]. Therefore, it focuses on some “interesting” events that have high prediction variance. One classical method for active learning is based on Query-By-Committee (QBC) that indirectly measures the variance for query candidates. For example, QBC first samples from the classifier’s parameter distribution based on the training data to create a “committee” of classifier variants. Then, the data is randomly sampled via the posterior distribution $P(y|x, \theta, \pi)$. The parameters $\hat{\theta}^k (k = \{1, 2, \cdots, K\}$, where $K$ is the total number of sampled parameter vectors) are estimated with the sampled data. With these parameters, we obtain $K$ classifiers, known as committee members, each of whom generates a label for the unlabeled events. Then, QBC can generate a disagreement of these committees that represents the prediction variance, or “interesting” event. Therefore, events with high variance are requested for labels by active learning either from experts or simulation tool.

### V. Numerical Results

To validate our results, we use the simulation tool of Positive Sequence Load Flow (PSLF) software with high-grade dynamic simulations [24].

For this paper, we conduct experiments extensively for validation. When running PSLF, we consider faults such as line trip, three-phase short circuit, and single-line to ground fault, etc. The Illinois 200-bus system, known as ACTIVSg200 case, is utilized to demonstrate the result. The network topology of the test case is shown in Fig. 3, where the red nodes are load nodes and the green nodes are generator nodes. The size of nodes represents their corresponding outputs or consumptions.

For PMU data, Fig. 4 shows the simulated PMU measurements from 3 types of events: line trip, single line to ground, and three-phase short circuits. Specifically, we implement these 3 events at time $t = 7s$ for line (5, 64) and clear them at time $t = 7.08s$. Normally, the sampling rate for PMU is 10 ∼ 60 samples per second, and therefore, we extract 1 sample every 0.02s. From Fig. 4, we find that there are distinguished dynamic behaviors for the 3 events.

### A. Feature Extraction

In this subsection, we try to validate our proposed feature selection method. For example, we conduct different events
to test the robustness of our method. Fig. 5 show a typical result with voltage magnitude, voltage angle, and frequency. Specifically, we calculate these features’ mean square distance of 50 points before and after an event. We can find that the distance shows large values at specific nodes, e.g., node 64, for voltage magnitude and angle. Therefore, our proposed feature selection will indicate voltage magnitudes and angles at node 64 to be selected.

B. Comparison of the Hybrid Approach with Other Supervised Learning Methods

In this subsection, we show the benchmark of supervised learning with our hybrid machine learning process. Specifically, we build the learning model as a multi-label classification model. In one case, 6 line trips are conducted, namely the lines (25, 64), (42, 44), (44, 200), (172, 180) and (172, 199). All the experiments are conducted under multiple loading conditions to mimic the reality.

To test the robustness of our supervised learning method against PMU number, we consider different PMU penetration levels of the grid. For example, 50% on the x-coordinate of Fig. 6 means that half of the buses are equipped with PMUs. For each penetration level, we randomly select the PMUs and use the selected PMU variables to create a feature pool. Such pool is then used to conduct our feature selection. After the feature extraction, the final features are vectorized into one sample vector. We collect 100 vectors for each event and train them through 5 machine learning methods and our hybrid method.

For coding, the experiment above is implemented for 10 times for each PMU penetration level to test the robustness of these methods. We plot the average accuracy of 5 machine learning methods and our hybrid machine learning method with respect to the penetration level of PMUs. The result is shown in Fig. 6. We can find that the accuracy varies among different machine learning methods and the penetration level of PMUs does not affect too much on different machine learning methods. Furthermore, our proposed hybrid machine learning method outperforms other supervised learning methods and its entropy based variance is relatively small under different PMU penetration levels. This result shows that our hybrid machine learning method successfully reduces the biases from different leaner and provides a confidence index for system operators.

C. Enhancement via Semi-supervised Learning and Active Learning

Based on the benchmark in V-B, we test if semi-supervised learning and active learning can be beneficial. Specifically, we fix the training samples of all the labels to be 100 in the training data set. For the testing data set, we let each label have 500 samples. Then, we gradually add the training samples of the 6 event classes clarified in Subsection V-B without any labels. The added number of unlabeled samples rises from 0 to 70. For each increment, we employ the Python package “sklearn” to implement semi-supervised learning and active learning on our training data. We compare the accuracy only on the testing data set for supervised, semi-supervised and active learning in Fig. 7.

We observe that firstly, semi-supervised learning and active learning make use of the information from unlabeled data so that the trained classifiers are more robust and accurate. When the number of unlabeled events in the training pool increases,
active learning to make the best use of unlabeled data in the faces challenges of not enough labeled data in the utility ability for the trained classifier. However, supervised learning based on selected features, we propose the flowchart of hybrid supervised learning and active learning for improving system event detection and identification. In this paper, we propose the system operators, is real-time situational awareness like been recognized in the industry to be of significant value to process to directly label those most unclear events.

Therefore, active learning converges to higher accuracy faster since it has a query process to directly label those most unclear events.

VI. CONCLUSION

One of the major synchrophasor applications that has been recognized in the industry to be of significant value to the system operators, is real-time situational awareness like event detection and identification. In this paper, we propose a framework to systematically design feature, conduct hybrid machine learning with confidence index, and use semi-supervised learning and active learning for improving system performance. In feature selection, we implement the wrapper method with a physical-knowledge-embedded metric design. Based on selected features, we propose the flowchart of hybrid supervised learning to reduce bias and increase generalization ability for the trained classifier. However, supervised learning faces challenges of not enough labeled data in the utility log. Therefore, we introduce semi-supervised learning and active learning to make the best use of unlabeled data in the PMU data stream. Further, simulations from PSLF and Python packages illustrate the effectiveness of the proposed methods.

REFERENCES