A Machine Learning Approach to Meter Placement for Power Quality Estimation in Smart Grid

Sardar Ali, *Student Member, IEEE*, Kui Wu, *Senior Member, IEEE*, Kyle Weston, and Dimitri Marinakis

***Abstract*—Due to the high-measuring cost, the monitoring of power quality (PQ) is nontrivial. This paper is aimed at reducing the cost of PQ monitoring in power network. Using a real-world PQ dataset, this paper adopts a learn-from-data approach to obtain a device latent feature model, which captures the device behavior as a PQ transition function. With the latent feature model, the power network could be modeled, in analogy, as a data-driven network, which presents the opportunity to use the well-investigated network monitoring and data estimation algorithms to solve the network quality monitoring problem in power grid. Based on this network model, algorithms are proposed to intelligently place measurement devices on suitable power links to reduce the uncertainty of PQ estimation on unmonitored power links. The meter placement algorithms use entropy-based measurements and Bayesian network models to identify the most suitable power links for PQ meter placement. Evaluation results on various simulated networks including IEEE distribution test feeder system show that the meter placement solution is efficient, and has the potential to significantly reduce the uncertainty of PQ values on unmonitored power links.**

***Index Terms*—Bayesian networks (BNs), conditional entropy (CE), Monte Carlo (MC) simulations, power quality (PQ) monitoring.**

Nomenclature

$f(d)$	Device transition function for a device d.
$c_i^{(d)}$	Power quality (PQ) of class i at device d.
C_i	Set of $c_i^{(d)}$ $\forall$ d.
$p^{(d)}_{c_y\|c_x}$	Probability that c_x will be mapped to c_y at device d.
$\widehat{d}$	Parent node of a node d.
$\check{d}$	Child node of a node d.
d_i	Inferred device.
d_o	Observed device.
F	Conditional transition function of device d_o given d_i.
$l^{(d)}_{\text{out}}$	Output link of device d.
$l^{(d)}_{\text{in}}$	Input link of device d.

I. Introduction

ELECTRICAL power networks are one of the critical infrastructures of our society. Due to our high dependence on electricity, the reliability in electric networks has become a core research interest in the smart grid area [1]. Reliability evaluation of power grid, however, is challenging [2]. One of the most influential factors impacting the reliability and energy saving of power networks is the PQ delivered to, and experienced by, critical electric equipment. Poor PQ, such as voltage sags, may lead to power outage and service interruptions. Hence, the monitoring of PQ is a crucial component of assessing and maintaining reliability in power grid.

Monitoring PQ, however, is not an easy task. Since the PQ measurement devices [3] are expensive, it is financially impractical to monitor every segment of a power network. The overhead of interconnecting these power meters and developing the power management system further increases the cost. Therefore, we need to intelligently place PQ meters on selected power links to reduce the uncertainty of PQ estimation on unmonitored links in the power grid. The following core challenge needs to address: given a fixed number of available power meters, which grid segments should be selected for monitoring such that PQ can be inferred in the remaining unmonitored segments of the network.

As the first step to tackle the above challenge, the probabilistic calculation of PQ values on unmonitored links requires the behavior (latent feature) of each device to be known. We represent the latent feature of a device as a transition function which is usually estimated through physical modeling or through the assessment of historical power monitoring data. Using a real-PQ dataset, we show that historical data can be used to capture the latent features of a device.

With devices' latent features captured, we in the second step introduce a network model which represents the smart microgrid as a data-driven network. In analogy, we represent the electrical components as network nodes, power links as data links, and flow of power as data flow on the links. This problem transformation significantly simplifies the complexity of the power network; it also presents the opportunity to use the well-investigated network monitoring and data estimation algorithms to solve the network quality monitoring problem.

Finally, we solve the intelligent meter placement problem by proposing an iterative approach for identifying network segments suitable for power meter placement. During each iteration of the algorithm, we identify in a greedy manner the network segment whose PQ is most unpredictable given the meters placed so far. We then place the next power meter

Manuscript received October 29, 2014; revised March 17, 2015; accepted June 1, 2015. This work was supported in part by Schneider Electric, and in part by the Natural Sciences and Engineering Research Council of Canada under Grant CRDPJ 428328-11. Paper no. TSG-01076-2014.

The authors are with the Department of Computer Science, University of Victoria, Victoria, BC V8W 2Y2, Canada (e-mail: sali@uvic.ca; wkui@ieee.org; kweston@kinsolresearch.com; dmarinak@kinsolresearch.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2015.2442837

at that location. In summary, this paper makes the following contributions.

1) A network model for PQ estimation, based on the device latent features that are learned from a real-world dataset.
2) An intelligent entropy-based algorithm and a Bayesian network (BN)-based approach to solve the meter placement problem.

II. Related Work

This paper is related to four categories of research and development: 1) PQ classification; 2) power reliability; 3) PQ improvement/estimation; and 4) meter placement.

On the first aspect, there are many approaches to the problem of classifying PQ events. Typically, quality is assigned a label based on the magnitude and duration of a voltage sag or swell. Electrical utilities typically report on indices such as system average root-mean-square (RMS) variation frequency index which is essentially a count of the number of times the magnitude and duration fall below a threshold. The IEEE also has a standard for classifying individual PQ events [4]. We use a discrete classification system in this paper, similar to that described in the IEEE Standard [4].

Regarding the second category, the industrial practices for electric power reliability in networks focus on measures such as mean time between failure, reliability, and availability as defined by the IEEE Gold Book [5]. The measures defined in [5] are theoretical values, measured or calculated for components and networks operating under normal conditions. They serve as methods for comparison but are not intended as predictive tools for networks that operate in realistic environments with varying load and PQ. It is known that there exists a relationship between PQ and the lifetime and performance of devices [6]. For an effective evaluation of power reliability, we need to accurately estimate PQ, which motivates the meter placement, and PQ estimation problems studied in this paper.

There have been recent studies to improve the electric PQ. In [7], a proactive approach was introduced to identify bad PQ events before they become a concern to end-users. The approach determines voltage threshold limits to determine if a potential voltage problem exists. Another recent study [8] uses genetic algorithm to estimate the harmonic states of the power network. The methodology was shown to be effective for estimating voltage and current state variables. A secondary control scheme is proposed in [9] to enhance the voltage quality of sensitive load bus in microgrids. Reference [10] proposed a transient state estimator to detect losses due to poor PQ. The estimator was validated on a test system to detect the presence of voltage sag/dip. Another estimator was proposed in [11] that improves the power consistency by identifying angle biases and current scaling errors using phasor-measurement-based state estimator.

On the fourth aspect, there is a great body of work on the problem of optimal sensor placement [12]. In the context of power networks, optimal placement of phasor measurement units (PMU) has been studied [13]. Reference [14] shows that adding few extra PMUs could improve the bad data detection in the network state estimation. A relevant work addressing the problem of distribution system state estimation was proposed [15] to minimize the state estimation errors. The optimal PMU placement and its communication infrastructure was designed [16] to address the problem of state estimation. A procedure finding the optimal trade-offs between PMUs and metering devices for distribution state estimation was investigated in [17]. Nevertheless, we have not seen any work on studying optimal meter placement problem in the context of network-wide PQ estimation. Further, there are three major differences between the existing PMU placement algorithms and our algorithm.

1) We focus on distribution networks at the enterprise level (e.g., a university campus).
2) Our method is data driven and is based on statistical machine learning method.
3) The existing PMU placement algorithms address the problem of estimating network states and do not consider PQ estimation explicitly.

III. Latent Features of Electric Devices

The objective of this paper is to reduce the cost of PQ monitoring by intelligently placing PQ meters on selected links in the power network. After deploying limited number of meters, we should be able to estimate the PQ values on unmonitored links as accurately as possible. A candidate link for meter placement is the one whose PQ is the most uncertain. The challenge here is how to identify the most uncertain links. Clearly, the PQ values on any power link is dependent on the physical characteristics of the electric devices. For example, the PQ at the output link of a UPS is more predictable than that of a switch. Hence, we need to know the behavior of each device in the network. We call the behavior of a device its latent feature or simply a transition function, which is usually estimated through physical modeling or through the assessment of historical power monitoring data.

In this section, we first introduce a latent feature model to capture the behavior of electric devices in the power network. Using a real-PQ dataset, we then demonstrate that the historical data can be used to capture the latent features of a device. We use k-fold cross-validation technique to measure the accuracy of latent features we obtain using our dataset. Experimental evaluations show that the captured latent features are consistent. The latent features (or transition functions) are then used to propose our meter placement algorithms.

A. Latent Feature Model

The latent feature model is basically a mechanism for capturing and mathematically representing the behavior of a device. We capture this behavior by monitoring the input and output links of electric devices and representing it as a transition function. A transition function $f(d)$ of a device d is the matrix consisting of real values representing the probabilities that a PQ input c_x is mapped to another PQ c_y at the output link of a device d. Fig. 1 shows the proposed latent feature model we use to capture $f(d)$. We use a real-world PQ dataset collected for a period of over four years to capture the latent features of various electric devices.

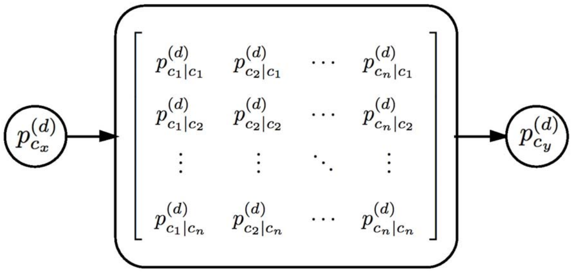
$p^{(d)}_{c_x}$
$p^{(d)}_{c_1|c_1}$ $p^{(d)}_{c_2|c_1}$ $\cdots$ $p^{(d)}_{c_n|c_1}$
$p^{(d)}_{c_1|c_2}$ $p^{(d)}_{c_2|c_2}$ $\cdots$ $p^{(d)}_{c_n|c_2}$
$\vdots$ $\vdots$ $\ddots$ $\vdots$
$p^{(d)}_{c_1|c_n}$ $p^{(d)}_{c_2|c_n}$ $\cdots$ $p^{(d)}_{c_n|c_n}$
$p^{(d)}_{c_y}$

Fig. 1. Latent feature model of a device d where the two circles represent the PQ meters at input and output of node d; the matrix inside the node d represents the transition function of the node.

B. Power Quality Dataset

Our PQ dataset was collected at an enterprise power network for a period of four years. For privacy and security reasons, the physical network structure/diagram is omitted. Instead, we represent the topology/positions of the installed PQ meters via a graph network as shown in Fig. 2. There are a total of ten PQ meters (numbered from m_1 to m_{10}) installed. Each meter reported the PQ events (sag/swell, transient, etc.) to the data collection server via Ethernet network. It is important to mention that we currently do not consider transmission network, but only focus on power distribution network at the enterprise-level, e.g., university campus. Hence, we collect the PQ dataset at an enterprise network located at the distribution level. The network is using a standard three-phase distribution system. Devices of varying loads are using this network, including electric vehicles and large motors. Three-phase transformers with four-wire output are used for 120 V service. Table I shows the number of events reported by each PQ meter while the positions of the meters are shown in Fig. 2.

The original PQ events reported by our PQ meters carry detailed information where some of the reported attributes are not directly relevant to PQ monitoring. For instance, we have a large number of branch circuit monitors installed that log every 15 min. Second, due to the detailed information content, the size of the raw dataset was about 40 GB. In order to simplify the format and make the dataset concise and easy to analyze, we transform the reported events into a tabular form consisting of the PQ attributes we used. As a result, there are about 6000 PQ events recorded in the dataset. Sample events from the dataset are shown in Table III.

1) Each row in the table represents a PQ event.
2) The magnitude field represents a percentage of the nominal voltage that the sag or swell reached at its maximum (for instance, the number 84 means that voltage is sagged to 84% of its nominal value and 147 means that it swelled up by 47% over its nominal value).
3) The severity field is a calculated statistic that combines the magnitude, duration and class of an event to provide a ranking variable.

Using IEEE Standard 1159 [4], we classify the PQ events based on the fluctuation of the voltage for a predefined period. There are 14 different PQ classes defined in the standard, denoted from c_1 to c_{14}, respectively. Table IV shows samples of the events we classify using the IEEE Standard where the PQ class is shown in the last column of the table. The frequency of events belonging to the IEEE PQ class (c_1–c_{14}) is shown in Table II.

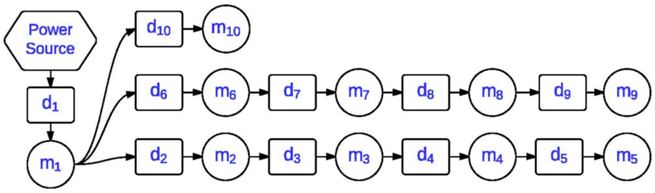

Fig. 2. Graph network of PQ meters installed in a power network.

C. Capturing the Latent Feature/Transition Function *f(d)*

Using the real-world PQ dataset, we capture the device latent feature in three simple steps as follows.

1) *Synchronizing the PQ Events:* The PQ meters in our data collection network were configured to report only bad PQ events. We noticed that, in some cases, there are bad PQ events reported at some links while nothing reported by other meters at that time instance. This happens when a device, for instance a UPS, maps a bad quality to good quality. In such cases, we assume a nominal PQ value (PQ class c_{14}) at the monitored but unreported points.
2) *Building Frequency Tables:* We now put all the PQ events in a 2-D array $M(i, j)$ of events where the first dimension of the array represents an event i in the time series while the second dimension represents the corresponding event for each device j. We then count the input to output PQ mappings at each device. This results in a 14×14 frequency table $fr(d)$ for each device d. As an example, frequency table for device d_8 is shown in Table V.
3) *Frequency to Probability Mapping:* Finally, the transition function is calculated by dividing every element of the frequency table $fr(d)$ by the sum of the row containing that element, i.e., $f(d, i, j) = fr(d, i, j)/\sum_{k=1}^{14} fr(d, i, k)$. Here, we slightly abuse the notation by using $f(d, i, j)$ to represent the value at the intersection of the ith row and the jth column in matrix $f(d)$. Hence, the transition function is represented with a matrix. If every element in a row (say ith row) of the frequency table is a 0, we assume the same probability (i.e., 1/14) for each element in that row in the transition function, implying that no knowledge can be learned from the dataset about the corresponding input event (c_i) on this device, and as such we assume the maximum uncertainty on its output events to avoid biased estimation. Table VI shows a sample transition function formulated from Table V.

D. Cross-Validation of $f(d)$

We use k-fold cross-validation technique to measure the accuracy of latent features we learned. We partition the dataset into k random samples of equal size. Out of the k samples, we use $k - 1$ samples to generate a training transition function and one sample to generate the test transition function. The cross-validation is repeated k times where each of the

TABLE I
FREQUENCY TABLE SHOWING THE NUMBER OF EVENTS GENERATED/REPORTED BY EACH PQ METER

Meter ID	1	2	3	4	5	6	7	8	9	10
No. of Events	1705	629	756	764	777	282	309	181	44	657

TABLE II
FREQUENCY TABLE SHOWING THE NUMBER OF EVENTS CLASSIFIED AS IEEE PQ CLASS (c_i)

Power Quality Class	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}
Number of Events	3056	738	1485	274	144	354	10	11	0	2	8	2	19	1

TABLE III
SAMPLE EVENTS FROM THE DATASET COLLECTED

Event ID	Node ID	Duration (seconds)	Magnitude (volts)	Severity	Type
119	5	0.02	292	3.19	Transient
338	6	1.002	147	47.1	Swell
763	1	0.07	84.4	1.03	Sag

TABLE IV
SAMPLE EVENTS CLASSIFICATION USING IEEE STANDARD 1159 [4]

Node ID	Duration (seconds)	Magnitude (volts)	IEEE Event Class
4	0.00065	127	c_1
4	0.00754	146	c_2
8	0.00013	132	c_1
7	0.049	84	c_3
6	1.002	147	c_7
6	0.518	79	c_6
1	0.001	131	c_1
7	0.016	82	c_3
6	7105.48	30	c_{12}
3	0.01664	233	c_4

TABLE V
SAMPLE FREQUENCY TABLE SHOWING THE NUMBER OF EVENTS MAPPED FROM INPUT PQ c_i TO OUTPUT PQ c_j AT DEVICE d_8

		Output PQ (c_j)					
		1	2	3	4	6	14
Input PQ (c_i)	3	4	16	4	2	0	113
	5	0	0	0	0	0	1
	6	0	0	0	0	5	32
	7	0	0	0	0	0	2
	12	1	0	0	0	0	0
	14	47	48	13	24	2	2122

TABLE VI
SAMPLE TRANSITION FUNCTION CAPTURED AT DEVICE d_8. ROWS AND COLUMNS HAVING ALL VALUES SET TO 0 ARE OMITTED

		Output PQ (c_j)					
		1	2	3	4	6	14
Input PQ (c_i)	3	0.03	0.12	0.03	0.01	0	0.81
	5	0	0	0	0	0	1.00
	6	0	0	0	0	0.14	0.86
	7	0	0	0	0	0	1.00
	12	1.00	0	0	0	0	0
	14	0.02	0.02	0.01	0.01	0	0.94

TABLE VII
MSEs IN ESTIMATED AND EXPECTED PROBABILITIES OF THE TRANSITION FUNCTIONS. STANDARD DEVIATION IN PQ VALUES OF THE k-FOLD TEST DATA

		k-fold cross-validation							
		Mean Square Error				Standard Deviation			
		2	10	100	500	2	10	100	500
Device (d_j)	2	0.008	0.012	0.023	0.027	4.37	4.59	4.99	5.04
	3	0.012	0.014	0.028	0.032	4.61	4.51	4.86	5.86
	4	0.014	0.017	0.031	0.036	4.65	4.73	4.82	5.04
	5	0.011	0.013	0.027	0.032	4.71	4.81	5.73	5.77
	6	0.007	0.010	0.021	0.025	2.62	2.89	3.07	4.47
	7	0.024	0.019	0.024	0.026	2.78	3.13	3.43	4.47
	8	0.017	0.015	0.021	0.025	2.38	2.57	3.35	4.59
	9	0	0.002	0.017	0.024	0.89	1.06	1.69	3.89
	10	0.007	0.01	0.021	0.026	3.66	3.72	4.05	5.26

k-samples is used exactly once for validation. The k results are then averaged to produce a single estimation for each device.

The mean-square error (MSE) is used to measure the variation of the validation/test function [represented as $f_v(d)$] from its training function [represented as $f_t(d)$]. The MSE is calculated as

$$\text{mse} = \sum_{i=1}^{14} \sum_{j=1}^{14} \mid f_v(d, i, j) - f_t(d, i, j) \mid / (i \times j).$$

We validate the latent features of all devices on various sample sizes. The largest training sample size is at $k = 2$ where we divide the entire dataset in two subsets of equal size; in this case, one subset is used to train the model while the other is used for validation. At the other extreme, at $k = 500$, the dataset is divided into 500 subsets where one of the subsets is used for validation while all other subsets are used for training.

Table VII shows the MSEs for all devices in the network with k-fold cross validation, where k is set to be different values. For each k-fold cross validation test, we also calculated the standard deviation of the k test results. It can be seen that when the value of k increases, the MSEs remain relatively stable with minor changes, but the standard deviation becomes larger. This is reasonable. When k increases, the number of samples in the test dataset becomes smaller, and the transition function built with a small number of samples in the test dataset becomes less accurate and leads to large variance in the test results. Nevertheless, the MSEs together with the standard deviation indicate that the test results with different k values do not exhibit significant statistical differences, and the small MSE values suggest that a device behavior (latent feature) can be captured accurately with historical PQ data from PQ meters.

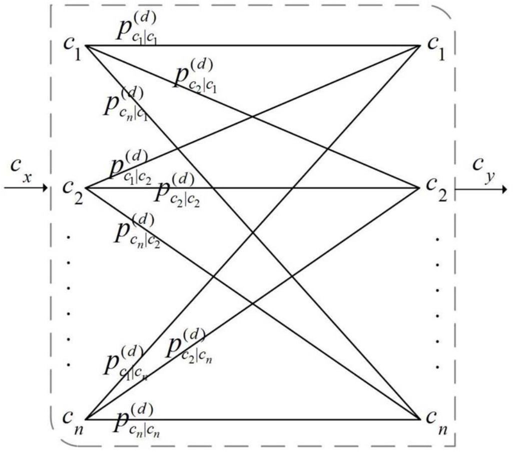

Fig. 3. PQ transition at each device d as a channel.

IV. Microgrid as Data Driven Network

We model the power network as a data-driven network, in analogy, where we represent the electrical components as network nodes, power links as data links, and the flow of power as data flow on the links. We assign the PQ on a link at an instance in time as a discrete class (from c_1 to c_n). Aligning with the meters' sampling interval, the time is slotted, and in every time slot, we record a PQ class of each link where a PQ meter is installed.

Moreover, in order to simplify our model, we treat the power flow through each node as a channel (shown in Fig. 3). The input and output of this channel at each node comprises n PQ classes. The probability that a PQ c_x will be "received" as c_y at the output of the channel at each device d is represented by the symbol $p^{(d)}_{c_y|c_x}$. For each device d, we call the $n \times n$ matrix consisting of the probability values $p^{(d)}_{c_y|c_x}$ the PQ transition function, or simply transition function. For a device (subnet) having multiple inputs/outputs, a PQ transition function is associated with each input/output pair. We represent the PQ transition function $f(d)$ of a device d as a matrix as

$$f(d) = \begin{bmatrix} p^{(d)}_{c_1|c_1} & p^{(d)}_{c_2|c_1} & \cdots & p^{(d)}_{c_n|c_1} \\ p^{(d)}_{c_1|c_2} & p^{(d)}_{c_2|c_2} & \cdots & p^{(d)}_{c_n|c_2} \\ \vdots & \vdots & \ddots & \vdots \\ p^{(d)}_{c_1|c_n} & p^{(d)}_{c_2|c_n} & \cdots & p^{(d)}_{c_n|c_n} \end{bmatrix} \quad (1)$$

where $p^{(d)}_{c_y|c_x}$ is the probability that the input quality c_x is received as c_y at the output of device d. Note that every row in the above matrix should sum to 1.

V. Intelligent Meter Placement

A. Bayesian Network-Based Approach

This section describes a BN-based algorithm for selecting locations for placing power meters in a power grid. The approach uses Monte Carlo (MC) sampling and probabilistic inference approaches to identify locations in the power grid which exhibit unpredictable PQ events.

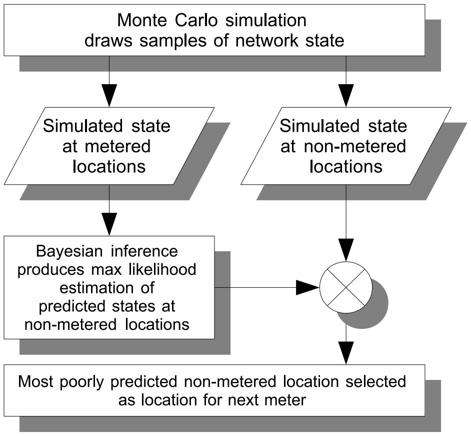
Monte Carlo simulation draws samples of network state
Simulated state at metered locations
Simulated state at non-metered locations
Bayesian inference produces max likelihood estimation of predicted states at non-metered locations
Most poorly predicted non-metered location selected as location for next meter

Fig. 4. Data flow diagram of meter selection process during a single iteration of the greedy algorithm.

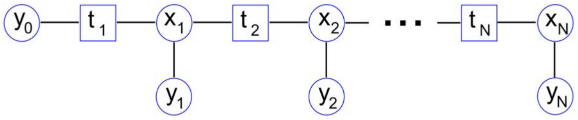

Fig. 5. Power network modeled as a factor graph.

The problem is inherently challenging as the information received from a power meter flows not only the forward direction from the root nodes toward the leaf nodes, but also in reverse or upstream direction toward the root node (utility main) and back to all other nodes in the network.

To tackle the above challenge, we cast the problem as a BN and model the power grid using a factor graph. Several message passing algorithms could be used to help us determine the optimal meter placement. We chose the belief propagation or sum-product algorithm [18], since it is well understood and has been shown to work for general topologies [19] including tree networks.

1) MC Event Sampling: Given the transition function $f(d)$, we use an MC method to obtain a set of K samples at each node d. We first compute a pmf $f_x(d)$ for each node d using its transition function $f(d)$ and the pmf of its parent node $\widehat{d}$ as $f_x(d) = f(d) \times f_x(\widehat{d})$. Then, at each time slot $i \in \{1 \ldots K\}$, we draw a sample $c^{(d)}_i$ from $f_x(d)$ at each node d. We repeat this at each node of the tree starting from the root and ending at the leaves. The result is a set of K simulated samples $C_i = \{c^{(1)}_i, c^{(2)}_i, \ldots, c^{(N)}_i\}$ for each of the N links in the power network.

2) Event Inference Using Belief Propagation: The samples obtained by the MC simulation of PQ propagation contain consistent sets of PQ values at both metered and nonmetered locations. We use Bayesian inference to infer the PQ at nonmetered locations as a function of the simulated values observed at the metered locations and compare the resulting predictions to the simulated value seen at the nonmetered locations. This process gives a relative indication of the predictive strength on each link of the network. Fig. 4 shows a high-level description of this process.

To do the prediction, we first model the power network as a factor graph (Fig. 5) and then use belief propagation to find

Algorithm 1: MC Predicted Error Algorithm

```
Input: The topology $T$ of the power grid; the $pmf$ of the input
       feed to first device i.e., $f_x(0)$; transition functions $f(d)$;
       maximum number of meters $M_{max}$; maximum
       uncertainty to allow $\delta_{max}$; and the number of
       Monte Carlo samples $K$ to draw
Output: $L$ (list of links to be selected for meter placement)
begin
    $M \leftarrow 0$; $\delta_{curr} \leftarrow \delta_{max}$;
    while ($M < M_{max}$ and $\delta_{curr} \geq \delta_{max}$) do
        $\epsilon_l \leftarrow 0$, $\forall$ links $l \in T$;
        /* $\epsilon_l$ is prediction error at link $l$ */
        foreach (Monte Carlo Sample $k$) do
            $L \leftarrow$ set of metered links $\in T$;
            $L' \leftarrow$ set of non-metered links $\in T$;
            $\widehat{C}_k \leftarrow$ predictPowerQuality($L', L, T$);
            /* $\widehat{C}_k$ is the $k^{th}$ set of predicted power quality
            values while $C_k$ is the $k^{th}$ set of sampled values at
            all links*/
            foreach (link $l \in L'$) do
                if $\widehat{c}_k^{(l)} \neq c_k^{(l)}$ then
                    $\epsilon_l \leftarrow \epsilon_l + \frac{1}{K}$; /* add $\frac{1}{K}$ to predicted error */
                end
            end
        end
        $selectedLink \leftarrow \max(\mathcal{E}).position$;
        /* $\mathcal{E} = \{\epsilon_l\}$ i.e., set of $\epsilon_l$ $\forall$ $l$ */
        $L$.add($selectedLink$);
        $\delta_{curr} \leftarrow \max(\mathcal{E})$; $M \leftarrow M + 1$;
    end
end

function predictPowerQuality($L', L, T$) : $C_k$
begin
    init pmf $\Psi = \{\psi_l\}$, $\forall$ links $l \in T$;
    $\Psi' \leftarrow$ BeliefPropogation given evidence $L$
    foreach ( link $l \in L'$) do
        $c_k^{(l)} \leftarrow$ max probability power quality class inferred in $\psi'_l$;
    end
    $C_k = \{c_k^{(l)}\}$
end
```

the inferred values of PQ at the output of each node using the (simulated) evidence obtained from the power meters. The factor graph has conditional probability nodes t, equality nodes x, and evidence nodes y. The t nodes represent actual electrical devices with a known transition function. The x nodes represent wired connections on our network for which we have already obtained a set of samples using MC sampling. These nodes are constrained so that all edges connected to them are equal. The y nodes represent locations where a power meter could be placed. The nonmetered nodes are initialized to a uniform pmf and the metered nodes are set to a trivial pmf with a probability of 1 at the true PQ event and 0 everywhere else.

For each time slot t_i, we infer the maximum likelihood PQ event that would appear at each node given the current meter configuration. We then estimate the error rate for each node in the network. If the inferred event differs from the event given by the MC sample we add $1/K$ for that sample. At each round of the algorithm we greedily choose to place a meter at the node with the highest error rate. We terminate the algorithm when all meters have been placed.

The number of required PQ meters could be determined based on: 1) the available financial budget; and 2) the desired estimation accuracy. We consider both aspects. The proposed algorithms keep placing meters until either the maximum meter limit is exhausted or the desired estimation accuracy is achieved. The estimation accuracy is captured by the certainty of PQ values on network segments (see Algorithm 1 and Fig. 4 for further details).

B. Conditional Entropy-Based Approach

Since, the PQ values on network segments are dependent on each other, we exploit the idea of conditional entropy (CE) to propose another new algorithm. Further, this approach is much faster than the BN-based approach without compromising the accuracy. The idea here is to install each power meter under consideration on a network segment i which results in maximum reduction in overall network entropy. We consider all possible placement points for every meter to be placed and choose a link which reduce the network entropy at maximum. Note that a reduction in network entropy is the sum of entropy reduction on the underlying link i and all other links whose entropy is minimized/reduced in effect of meter placement on a segment i. The one time matrix multiplications in this approach are much faster than our previous requirement of resampling the network state after every possible meter placement.

The CE-based algorithm is efficient and scalable to large scale real-world networks. Both BN and CE approaches are based on similar concepts of predicting the state of PQ values at unmonitored links given the current network configuration (positions of meters already placed). The CE-based approach, which we will call MinEntropy, uses a heuristic to combine evidence but results in orders of magnitude faster running time.

1) Methodology: As discussed earlier in this paper, the uncertainty of PQ values on a link is dependent on the uncertainty of PQ values on other links (parents, children, sibling nodes, etc.) in the network. Therefore, any new information about PQ values at a link increase our belief of the PQ values on other dependent links in the same network. Technically, the entropy of any link in the network is reduced by an amount of ≥ 0 by knowing the values of PQ on any other link in the network. We also know that, the entropy of a link given another link is always less than or equal to its original entropy i.e., $H(Y \mid X) \leq H(Y)$. Since every link $l_{\text{out}}^{(d)}$, if chosen for meter placement, influences the uncertainty of PQ values on other links, we consider the CE of all monitored links while placing power meter at a link $l_{\text{out}}^{(d)}$.

Now, the CE of a link $l_{\text{out}}^{(d_i)}$ (the output link of the inferred device d_i) given the meter is being installed on a link $l_{\text{out}}^{(d_o)}$ (the output link of the device d_o) is calculated using the formula

$$H(Y \mid X) = \sum_{x \in X} \left(p(x) \sum_{y \in Y} p(y \mid x) \log\left(\frac{1}{p(y \mid x)} \right) \right)$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

where X and Y are the distribution functions of the output links of d_o and d_i, respectively. We write the above equation in terms of our PQ distribution vector $f_x(d_o)$, device transition matrix $f(d_i \mid d_o)$ as

$$H(Y \mid X) = -\sum (f_x(d_o) \times (F \otimes \log f(d_i)))$$

where × represents the cross product, the symbol $\otimes$ represents the dot or component-wise product (also known as Hadamard product), and log is a component-wise log operation. Further, the $\sum$ operation is the summation of components of the resulting vector after $\otimes$ and then $\times$ operations, and F is the conditional transition function representing $f(d_i \mid d_o)$. Depending on the positions of d_o and d_i, F is calculated in one of the three methods as follows.

1) *Observed Device d_o is a Parent of d_i:* Here, the conditional transition function $f(\text{child} \mid \text{parent})$ is simply the product of the normal transition functions of devices between links $l_{\text{out}}^{(d_o)}$ and $l_{\text{out}}^{(d_i)}$, that is

$$F = f(\check{d}_o) \times \ldots \times f(d_i).$$

2) *Observed Device d_o is a Child of d_i:* We calculate the influence of a child device on a parent device. Note that the parent may not necessarily be the immediate parent. To calculate the entropy of parent given child using the general formula of CE, we need to first calculate the conditional transition function F. We use the concept of posterior probability (the Bayes theorem) to calculate F. This function is simply the product of the reverse transition functions of devices all the way from child to parent. The reverse transition function $f'(d)$ (consist of $p(\text{parent} \mid \text{child})$ or $p(X \mid Y)$) is calculated as $p(X \mid Y) = (p(X)p(Y/X))/(p(Y))$. In our case, the function $f'(d)$ of a device d which list $p(x \mid y)$ in the xth row and yth column is calculated as

$$f'(d) = \begin{bmatrix} f_x(\widehat{d}) \\ f_x(\widehat{d}) \\ \vdots \\ f_x(\widehat{d}) \end{bmatrix} \otimes \left[f(d)\right]^T \oslash \begin{bmatrix} f_x(d) \\ f_x(d) \\ \vdots \\ f_x(d) \end{bmatrix}^T$$

where $\otimes$ is the component-wise product, $\oslash$ is the component-wise division, and $\widehat{d}$ is the immediate parent of device d. Finally

$$F = f'(d_o) \times f'\left(\widehat{d_o}\right) \times \cdots \times f'(\check{d}_i).$$

3) *Devices d_o, d_i Belong to Different Subtrees:* This is an interesting case where the devices d_o and d_i belong to two different sub-trees rooted by a device d_r. In this case, F is calculated in two steps. First, we calculate the conditional transition function $f(d_r \mid d_o)$ of devices between links $l_{\text{out}}^{(d_o)}$ and $l_{\text{out}}^{(d_r)}$ using method 2. We then calculate the conditional transition function $f(d_i \mid d_r)$ of devices between links $l_{\text{out}}^{(d_r)}$ and $l_{\text{out}}^{(d_i)}$ using method 1. Finally

$$F = f(d_i \mid d_o) = f(d_r \mid d_o) \times f(d_i \mid d_r).$$

Algorithm 2: MinEntropy Algorithm

```
Input: The topology $T$ of the power grid; the $pmf$ of the input
       feed to first device i.e., $f_x(0)$; transition functions $f(d)$;
       maximum number of meters $M_{max}$; and maximum
       uncertainty to allow $\delta_{max}$
Output: $L$ (list of devices to be selected for meter placement)
begin
    $M \leftarrow 0$; $\delta_{curr} \leftarrow \delta_{max}$;
    while ($M < M_{max}$ and $\delta_{curr} \geq \delta_{max}$) do
        $maxReduction \leftarrow 0$;
        $selectedLink \leftarrow 0$;
        foreach (device $d$ in $T$) do
            $entReduction \leftarrow$ calcNetworkEntropy($d$);
            if $maxReduction < entReduction$ then
                $maxReduction \leftarrow entReduction$;
                $selectedLink \leftarrow d.outputLink$;
            end
        end
        $L$.add($selectedLink$);
        updateEntopies($selectedLink$);
        $\delta_{curr} \leftarrow \max(\mathcal{E})$; $M \leftarrow M + 1$;
    end
end

function calcNetworkEntropy($d$) : $entRed$
begin
    F $\leftarrow$ identityMatrix(n);
    /* $F$ is the combined transition function i.e., $f(d_i \mid d_o)$ */
    $d_o \leftarrow d$; $d_p \leftarrow d$; $d_i \leftarrow d$;
    entRed $\leftarrow$ recursiveConditional($d_0, d_p, d_i, F$);
end

function recursiveConditional($d_o, d_p, d_i, F$) : entRed
begin
    condEnt $\leftarrow$ - sum ( $f(d_o) \times F \otimes \log(F)$);
    entRed $\leftarrow$ entropy($d_i$) $-$ condEnt;
    foreach (immediate child $c$ of $d_i$) do
        $F \leftarrow F \times f(c)$; /* child given parent link */
        /* for next recursive call, $c$ is the inferred device and
        $d_i$ is the previous device */
        $d_p \leftarrow d_i$;  $d_i \leftarrow c$;
        if ($d_i \neq d_p$) then
            entRed $\leftarrow$ entRed +
            recursiveConditional($d_o, d_p, d_i, F$);
        end
    end
    $d_p \leftarrow d_i$;  $d_i \leftarrow$ getParent($d_i$);
    if ($d_i \neq -1$ and $d_i \neq d_p$) then
        $F \leftarrow F \times f'(c)$; /* parent given child link */
        entRed $\leftarrow$ entRed + recursiveConditional($d_o, d_p, d_i, F$);
    end
end
```

2) MinEntropy Algorithm: Algorithm 2 illustrates our CE-based solution to power meter placement. The idea here is to install each meter under consideration on a link i of the network which results in maximum reduction in overall network entropy. We consider all possible placement points for every meter to be placed and choose a link that reduce the network entropy to a minimum. Note that a reduction in network entropy is the sum of entropy reduction on the underlying link i and all other links whose entropy is minimized/reduced in effect of meter placement on link i.

In order to calculate the network entropy for every candidate link $l_{\text{out}}^{(d_o)}$, we first calculate the entropy of every link $l_{\text{out}}^{(d_i)}$ given $l_{\text{out}}^{(d_o)}$. These conditional entropies are efficiently

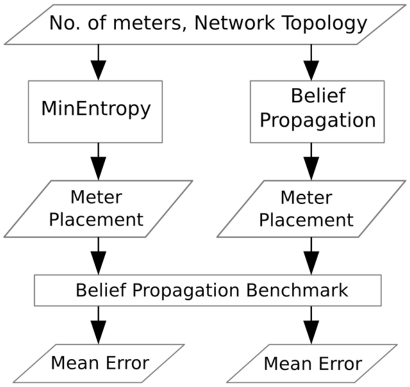

Fig. 6. Overview of the meter placement evaluation process.

TABLE VIII
EVENT TYPES

Type	Event Description
1	Good/normal power quality.
2	Below 70% of nominal voltage for more than 0.02 seconds or below 80% of nominal voltage for more than 0.5 seconds.
3	Below 70% of nominal voltage for more than 0.2 seconds.
4	Interruption of at least 1 second.
5	Interruption of at least 5 minutes.

TABLE IX
TRANSITION FUNCTIONS OF VARIOUS ELECTRICAL COMPONENTS OBTAINED USING OUR LATENT FEATURE MODEL. (a) BUS. (b) SWITCH. (c) TRANSFORMER. (d) UPS

(a)

		Output PQ				
		1	2	3	4	5
Input PQ	1	0.9	0.1	0	0	0
	2	0	0.9	0.1	0	0
	3	0	0	0.9	0.1	0
	4	0	0	0	0.9	0.1
	5	0	0	0	0	1

(b)

		Output PQ				
		1	2	3	4	5
Input PQ	1	0.7	0.1	0.1	0.05	0.05
	2	0	0.7	0.1	0.1	0.1
	3	0	0	0.7	0.2	0.1
	4	0	0	0	0.7	0.3
	5	0	0	0	0	1

(c)

		Output PQ				
		1	2	3	4	5
Input PQ	1	0.85	0.15	0	0	0
	2	0.15	0.7	0.15	0	0
	3	0	0.15	0.7	0.15	0
	4	0	0	0.15	0.7	0.15
	5	0	0	0	0	1

(d)

		Output PQ				
		1	2	3	4	5
Input PQ	1	1	0	0	0	0
	2	0.8	0.2	0	0	0
	3	0.8	0	0.2	0	0
	4	0.8	0	0	0.2	0
	5	0.8	0	0	0	0.2

calculated by multiplying $f_x(d_o)$ with transition functions of all devices on the path between links $l_{\text{out}}^{(d_o)}$ and $l_{\text{out}}^{(d_i)}$. We do not need to explicitly identify the path from $l_{\text{out}}^{(d_o)}$ to $l_{\text{out}}^{(d_i)}$ and we do not need to multiply the same transition functions again and again. The entropy calculation works in recursive fashion. Once we calculate the CE for a directly connecting neighbor of $l_{\text{out}}^{(d_o)}$, we then recursively calculate the entropies of neighboring links of that neighbor. Here, it should be noted that: a) every link trigger the neighboring links except the one who triggered the link itself. So no infinite recursion takes place and every link is accessed only once; b) the product of transition functions calculated from $l_{\text{out}}^{(d_o)}$ to some $l_{\text{out}}^{(d_k)}$ is used to calculate the next product; and c) if a link is invoking its parent link, we use reverse transition function $f'(d)$ of that device. Otherwise, the normal transition function $f(d)$ is used. After every meter placement, the link entropies are updated. The same process is repeated until all PQ meters are placed.

VI. EVALUATIONS

We evaluate the two algorithms on a set of simulated networks. The evaluation process is depicted in Fig. 6 where each algorithm is given the same network topology to place a set of M meters. The devices considered include bus, switch, transformer and UPS. PQ events are assigned a number from 1 to 5 in order of severity in accordance with [20] where the lower number represents a clean input. These are listed in Table VIII along with their descriptions. Table IX lists transition functions of various electrical components obtained from a real-world PQ dataset using our latent feature model. From the same dataset, we learn a prior on the utility feed of [0.9947 0.005 0.0002 0.00009 0.00001]. The algorithms we propose are generic and could be used for placing meters in other network configuration. We evaluate our algorithms on different topologies and network configurations including the IEEE 13-node distribution test feeder network. For the BP algorithm, we collect $N = 10\,000$ samples for each device using MC sampling. For each network configuration, we place $M = 5$ m in order of importance.

Fig. 7 shows the meters placed by the two algorithms in various network topologies. The positions of the meters placed by both algorithms are essentially similar. The MinEntropy algorithm achieves much faster results, completing in less than a second in all cases. On the other hand, the BP takes a longer time to complete. This is because BP compares individual samples on all links for every possible placement while the MinEntropy approach computes the conditional entropies at nonmetered locations using probability mass functions instead of using individual samples. Algorithm completion times for both BP and CE approaches are shown in Table X.

The meter placements are then passed to the belief propagation benchmark to compare the accuracy of the two algorithms in terms of MSE, i.e., the mean error between the estimated and actual transition functions on unmonitored links. We collect a set of known samples for a given meter configuration and infer the maximum likelihood PQ event that would appear at each nonmetered node using belief propagation. We then estimate the error rate for each node. If the inferred event differs from the event given by the MC sample, we add $1/N$ for that sample. The mean error rate across all nodes is taken as the final performance metric. As shown in Table X, the MSEs are very small for both algorithms in all networks we tested. The BP algorithm gave slightly better estimations than MinEntropy in some cases at a cost of longer running times.

We also compare the accuracy of our algorithms in terms of cost savings. Table XI shows the number of meters needed by each algorithm where it can be clearly seen that using our solution to achieve the same accuracy (error rate

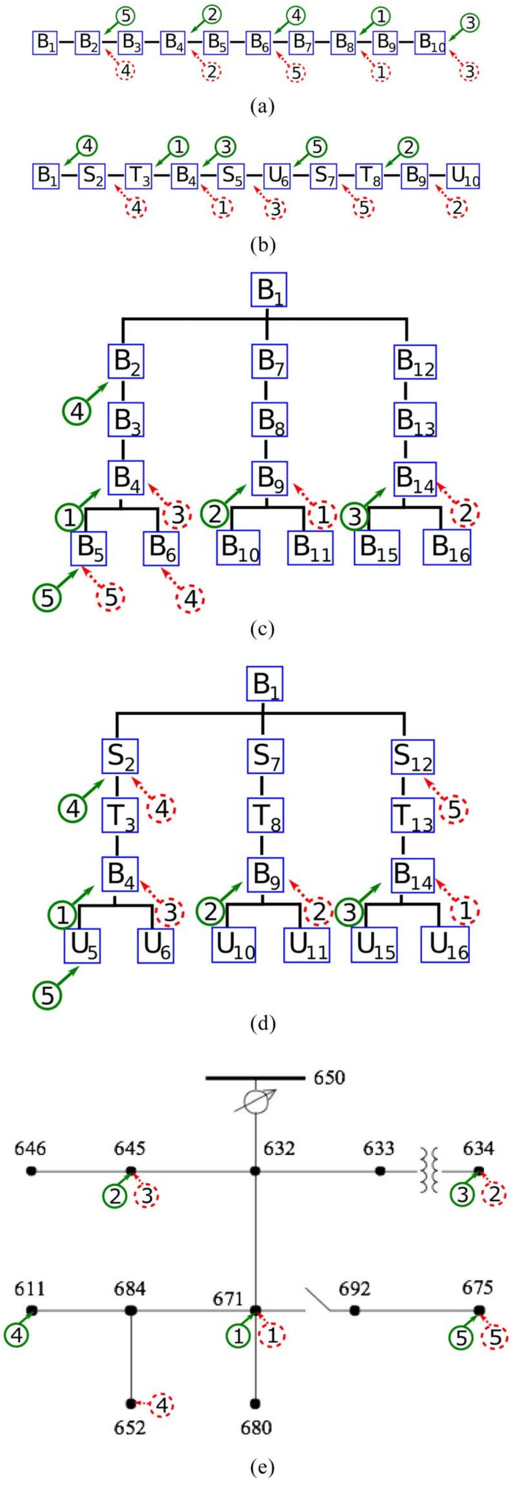
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10
(a)
B1 S2 T3 B4 S5 U6 S7 T8 B9 U10
(b)
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16
(c)
B1 S2 T3 B4 U5 U6 S7 T8 B9 U10 U11 S12 T13 B14 U15 U16
(d)
650
646
645
632
633
634
611
684
671
692
675
652
680
(e)
1
2
3
4
5

Fig. 7. Networks used in our experiments. B = bus, S = switch, T = transformer, and U = UPS. Ordered dotted circles correspond with the sequence of meters placed by BP while the solid circles show the meter placed by MinEntropy. (a) Homogenous line network. (b) Heterogeneous line network. (c) Homogeneous tree network. (d) Heterogeneous tree network. (e) IEEE 13-node distribution test feeder network.

of < 0.05) can reduce the number of meters by 33%. Results of the random placement algorithm were obtained by randomly placing meters until the desired accuracy was achieved.

TABLE X
RESULTS FOR EACH NETWORK CONFIGURATION

Network Topology	Algorithm	Meter Placement Sequence	Mean Error Rate	Elapsed Time (seconds)
Line homogeneous	BP	9,5,11,3,7	0.041112	270
	MinEnt	9,5,11,7,3	0.041112	0.064
Line heterogeneous	BP	5,10,6,3,8	0.040215	281
	MinEnt	4,9,5,2,7	0.064040	0.064
Tree homogeneous	BP	10,15,5,7,6	0.057893	727
	MinEnt	5,10,15,3,6	0.056891	0.138
Tree heterogeneous	BP	15,10,5,3,13	0.063510	727
	MinEnt	5,10,15,3,6	0.071655	0.138
IEEE 13-Node Test Feeder	BP	671,634,645,652,675	0.052381	802
	MinEnt	671,645,634,611,675	0.052381	0.138

TABLE XI
NUMBER OF METERS REQUIRED IN VARIOUS NETWORKS TO RESTRICT THE MEAN ERROR RATE TO 0.05 (5%)

Algorithm	Line Homogen.	Line Heterogen.	Tree Homogen.	Tree Heterogen.	IEEE 13-Node Test Feeder
BP	5	5	6	6	6
MinEnt	5	6	6	6	6
Random	7	8	8	8	8

VII. CONCLUSION

PQ meters are expensive devices and it is financially infeasible to install them on every link in the power network. We proposed algorithms which intelligently place power meters on selected power links to reduce the cost of PQ monitoring. We formulated the problem of selecting suitable meter placements in power networks such that PQ can be best predicted. Two approaches were presented, one based on CE and one considering prediction error. Experiments in various simulation networks including the IEEE 13-node distribution test feeder suggested that the CE-based MinEntropy approach is much faster. Finally, the proposed solutions significantly reduce the uncertainty of PQ values on unmonitored power links.

REFERENCES

[1] K. Moslehim and R. Kumar, “A reliability perspective of the smart grid,” *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 57–64, Jun. 2010.

[2] R. Albert, I. Albert, and G. L. Nakarado, “Structural vulnerability of the North American power grid,” *Phys. Rev. E*, vol. 69, no. 2, Feb. 2004, Art. ID 025103.

[3] Schneider-Electric. (Aug. 2013). *Smart Meters*. [Online]. Available: http://www.smartmeters.com/tag/schneider-electric.html

[4] *IEEE Recommended Practice for Monitoring Electric Power Quality*, IEEE Standard 1159, Jun. 2009.

[5] *IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems*, IEEE Standard 493, 2007.

[6] “ITI (CBEMA) curve application note,” Inf. Technol. Ind. Council, Washington, DC, USA, Tech. Rep. 3, Oct. 2007.

[7] K. D. McBee and M. G. Simoes, “Utilizing a smart grid monitoring system to improve voltage quality of customers,” *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 738–743, Jun. 2012.

[8] C. F. M. Almeida and N. Kagan, “Harmonic state estimation through optimal monitoring systems,” *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 467–478, Mar. 2013.

[9] M. Savaghebi, A. Jalilian, J. C. Vasquez, and J. M. Guerrero, “Secondary control for voltage quality enhancement in microgrids,” *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1893–1902, Dec. 2012.

[10] A. Farzanehrafat and N. R. Watson, “Power quality state estimator for smart distribution grids,” *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2183–2191, Aug. 2013.

[11] S. G. Ghiocel *et al.*, "Phasor-measurement-based state estimation for synchrophasor data quality improvement and power transfer interface monitoring," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 881–888, Mar. 2014.

[12] A. Krause and C. Guestrin, "Optimizing sensing: From water to the Web," *IEEE Comput. Mag.*, vol. 42, no. 8, pp. 38–45, Aug. 2009.

[13] W. Yuill, A. Edwards, S. Chowdhury, and S. P. Chowdhury, "Optimal PMU placement: A comprehensive literature review," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, San Diego, CA, USA, Jul. 2011, pp. 1–8.

[14] J. Chen and A. Abur, "Placement of PMUs to enable bad data detection in state estimation," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1608–1615, Nov. 2006.

[15] R. Singh, B. C. Pal, R. A. Jabr, and R. B. Vinter, "Meter placement for distribution system state estimation: An ordinal optimization approach," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2328–2335, Nov. 2011.

[16] M. Shahraeini, M. S. Ghazizadeh, and M. H. Javidi, "Co-optimal placement of measurement devices and their related communication infrastructure in wide area measurement systems," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 684–691, Jun. 2012.

[17] J. Liu *et al.*, "Trade-offs in PMU deployment for state estimation in active distribution grids," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 915–924, Jun. 2012.

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann, 1988.

[19] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 2001, pp. 689–695.

[20] *Analysis of Extremely Reliable Power Delivery Systems: A Proposal for Development and Application of Security, Quality, Reliability, and Availability (SQRA) Modeling for Optimizing Power System Configurations for the Digital Economy*, EPRI, Palo Alto, CA, USA, 2002.

Sardar Ali (S'11) received the B.S. degree from the Kohat University of Science and Technology (KUST), Kohat, Pakistan, and the M.S. degree from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2006 and 2009, respectively, both in information technology. He is currently pursuing the Ph.D. degree in computer science from the University of Victoria, Victoria, BC, Canada.

His current research interests include data analysis, information monitoring systems, smart microgrids, and Internet/network security.

Kui Wu (M'02–SM'07) received the B.Sc. and M.Sc. degrees from Wuhan University, Wuhan, China, in 1990 and 1993, respectively, and the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 2002, all in computer science.

He joined the Department of Computer Science, University of Victoria, Victoria, BC, Canada, in 2002, where he is currently a Professor. His current research interests include smart grid, network performance evaluation, and cloud computing.

Kyle Weston received the Diploma degree in electronics engineering technology from Camosun College, Victoria, BC, Canada, in 2003; the B.Eng. degree in electrical engineering from the University of Victoria, Victoria, in 2006; and the M.Eng. degree in intelligent systems from McGill University, Montreal, QC, Canada, in 2010.

He has been with Kinsol Research Inc., Victoria, for the past five years doing research and development for various software inference problems. His current research interests include computer vision, machine learning, and digital signal processing.

Dimitri Marinakis received the B.Sc. degree from the University of Victoria, Victoria, BC, Canada, in 1999, and the Ph.D. degree from McGill University, Montreal, QC, Canada, in 2009, both in computer science.

He is currently the Chief Scientific Officer at Kinsol Research Inc., Victoria. His current research interests include sensor networks, probabilistic reasoning, and machine learning.