A Machine Learning Approach to Power Quality Estimation in Smart Grid

Sardar Ali, Kui Wu, Kyle Weston, and Dimitri Marinakis
Department of Computer Science, University of Victoria, BC, Canada
sali@csc.uvic.ca, wkui@ieee.org, kweston@kinsolresearch.com, dmarinak@kinsolresearch.com

***Abstract*—Due to the high measuring cost, the monitoring of power quality is non-trivial. This paper is aimed at reducing the cost of power quality monitoring in power network. Using a real-world power quality dataset, the paper adopts a learn-from-data approach to obtain a device latent feature model, which captures the device behaviour as a power quality 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 power quality 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 power meter placement. Experimental results show that the meter placement solution is efficient, and has the potential to significantly reduce the uncertainty of power quality values on unmonitored power links.

***Index Terms*—Power Quality Monitoring, Monte Carlo Simulations, Bayesian Networks, Conditional Entropy**

NOMENCLATURE

$f(d)$	Device transfer function for a device d
$c_i^{(d)}$	Power quality of class i at device d
C_i	Set of $c_i^{(d)}\ \forall\ d$
$p_{c_y\|c_x}^{(d)}$	Probability that c_x will be mapped to c_y at device d
$\widehat{d}$	Parent node of a node d
$\breve{d}$	Child node of a node d
d_i	Inferred device
d_o	Observed device
F	Conditional transfer function of device d_o given d_i
$l_{out}^{(d)}$	Output link of device d
$l_{in}^{(d)}$	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 power quality delivered to, and experienced by, critical electric equipment. Poor power quality, such as voltage sags, may lead to power outage and service interruptions. Hence, the monitoring of power quality is a crucial component of assessing and maintaining reliability in power grid.

Monitoring power quality, however, is not an easy task. Since the power measurement devices [4] (termed as smart meters in this paper) 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 smart meters on selected power links to reduce the uncertainty of power quality 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 power quality can be inferred in the remaining unmonitored segments of the network.

As the first step to tackle the above challenge, the probabilistic calculation of power quality values on unmonitored links requires the behaviour (latent feature) of each device to be known. We represent the latent feature of a device as a transfer function which is usually estimated through physical modelling or through the assessment of historical power monitoring data. Using a real power quality 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 of the smart microgrid as a data-driven network, in analogy, where 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 power quality is most unpredictable given the meters placed so far. We then place the next power meter at that location. In summary, the paper makes the following contributions:

- A network model for power quality estimation, based on the device latent features that are learnt from a real-world dataset,
- An intelligent entropy-based algorithm and a Bayesian

network based approach to solve the meter placement problem.

II. Related Work

This work is related to three categories of research and development: power quality (PQ) classification, power reliability, and smart meter placement.

On the first aspect, there are many approaches to the problem of classifying power quality 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 RMS Variation Frequency Index (SARFI) 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 power quality events [5]. We use a discrete classification system in this work, similar to that described in the IEEE standard [5].

Regarding the second category, the industrial practices for electric power reliability in networks focus on measures such as Mean Time Between Failure (MTBF), reliability, and availability as defined by the IEEE Gold Book [6]. The measures defined in [6] 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 power quality. It is known that there exists a relationship between power quality and the lifetime and performance of devices [7]. For an effective evaluation of power reliability, we need to accurately estimate power quality, which motivates the meter placement, and power quality estimation problems studied in this paper.

On the third aspect, there is a great body of work on the problem of optimal sensor placement problem [8]. In the context of power networks, optimal placement of phasor measurement units (PMU) has been studied [9]. Nevertheless, we have not seen any work on studying optimal meter placement problem in the context of network-wide power quality estimation.

III. Latent Features of Electric Devices

Power quality meters are expensive devices, which need to place on carefully selected links in the power network. In order to estimate the power quality values on unmonitored links, we use the known power quality values from events reported by meters placed on monitored links. The probabilistic calculation of power quality values on unmonitored links requires the behaviour (latent feature) of each device to be known. We represent the latent feature of a device as a transfer function, which is usually estimated through physical modelling or through the assessment of historical power monitoring data. In this section, we first introduce a latent feature model to capture the behaviour of electric devices in the power network. Using a real power quality 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.

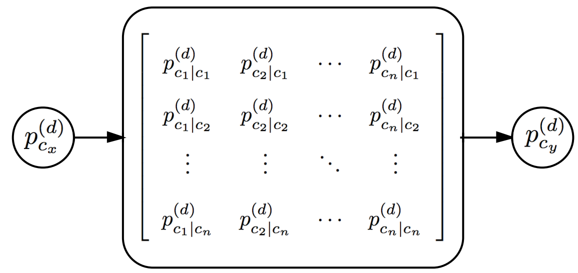


Fig. 1. Latent feature model of a device d where the two circles represent the power quality meters at input and output of node d; the matrix inside the node d represents the transfer function of the node.

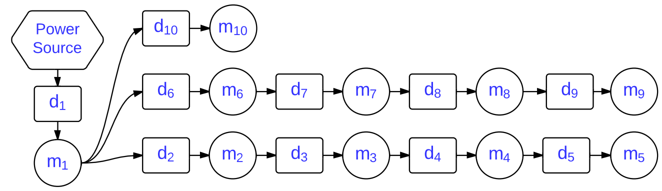


Fig. 2. Graph network of power quality meters installed in a power network.

A. The Latent Feature Model

The latent feature of a device is basically the behaviour of a device in the power network, which can be captured by monitoring the power quality at the input and output links of the device. Based on these power quality readings (events), we model a device behaviour as a transfer function $f(d)$. A transfer function of a device is the matrix consisting of real values representing the probabilities that a power quality input c_x is mapped to another power quality c_y at the output link of a device d. Figure 1 shows a sample node (shown as a square box) whose input and output links are monitored (using power quality meters shown in circles) to capture the latent feature $f(d)$. Once the latent feature $f(d)$ is known, we can probabilistically estimate the power quality at links where no power meter is installed.

B. Power Quality Dataset

Our power quality 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 smart meters via a graph network as shown in Fig. 2. There are a total of 10 power quality meters (numbered from m_1 to m_{10}) installed. Each meter reported the power quality events (sag/swell, transient, etc.) to the data collection server via ethernet network. Table I shows the number of events reported by each power quality meter while the positions of the meters are shown in Fig. 2.

The original power quality events reported by our power quality meters carry detailed information where some of the reported attributes are not directly relevant to power quality monitoring. For instance, we have a large number of branch circuit monitors installed that log every 15 minutes. 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

TABLE I
FREQUENCY TABLE SHOWING THE NUMBER OF EVENTS GENERATED/REPORTED BY EACH POWER QUALITY 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 POWER QUALITY 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.

ID	Source	Duration	Magnitude	Severity	Phase	Type
119	5	0.02	292	3.19	V1	Transient
338	6	1.002	147	47.1	V3	Swell
763	1	0.07	84.4	1.03	V3	Sag

TABLE IV
SAMPLE EVENTS CLASSIFICATION USING IEEE STANDARD 1159 [5].

Source	Start Time	Duration	Magnitude	IEEE Event Class
4	733051.9385	0.00065	127	c_1
4	733052.9522	0.00754	146	c_2
8	733452.0117	0.00013	132	c_1
7	733462.7471	0.049	84	c_3
6	733488.8235	1.002	147	c_7
6	733569.0525	0.518	79	c_6
1	733572.9232	0.001	131	c_1
7	733589.9307	0.016	82	c_3
6	733724.0312	7105.48	30	c_{12}
3	733724.1134	0.01664	233	c_4

TABLE V
A SAMPLE FREQUENCY TABLE SHOWING THE NUMBER OF EVENTS MAPPED FROM INPUT POWER QUALITY c_i TO OUTPUT POWER QUALITY 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

the reported events into a tabular form consisting of the power quality attributes we used. As a result, there are about 6000 power quality events recorded in the dataset. Sample events from the dataset are shown in Table III where: a) each row in the table represents a power quality event; b) 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, 147 means that it swelled up by 47% over its nominal value); c) 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 [5], we classify the power quality events based on the fluctuation of the voltage for a predefined period. There are 14 different power quality 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 power quality class is shown in the last column of the table. The frequency of events belonging to the IEEE power quality class (c_1 to c_{14}) is shown in Table II.

C. Learning Latent Feature ($f(d)$)

Using the real-world power quality dataset, we capture the device latent feature as follows:

- Since most of the PQ events are correlated, i.e., when there is a bad power quality event reported by a meter, the effect may be cascaded and a bad quality event may occur at other links (not necessarily at all links) as well. We identify these cascaded events based on timestamps of the events. When the time stamp is same (or varies by less than a second), we consider them correlated.
- For a cascaded event not occurring at a link, we set a nominal PQ value (PQ class c_{14}) at that link.
- We put all the events in a 2-dimensional 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.
- The latent feature of a device d is a function giving the probability of an output PQ value at that device given an input value. For every device, we count the frequency of PQ mappings of input to output PQ values. 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.
- Finally, the transfer function is calculated by dividing every element of the frequency table by 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 i-th row and the j-th column in matrix $f(d)$. Hence, the transfer function is represented with a matrix. If every element in a row (say i-th 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 transfer function, implying that no knowledge can be learnt 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 transfer 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 learnt. We partition the dataset

TABLE VI

A SAMPLE TRANSFER 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

MEAN SQUARE ERRORS (MSES) IN ESTIMATED AND EXPECTED PROBABILITIES OF THE TRANSFER FUNCTIONS.

		Subsets (k)						
		2	3	5	10	20	50	100
Device (d_j)	2	0.006	0.008	0.007	0.013	0.016	0.019	0.023
	3	0.011	0.011	0.014	0.016	0.02	0.025	0.028
	4	0.015	0.014	0.016	0.019	0.021	0.028	0.031
	5	0.01	0.01	0.012	0.015	0.018	0.023	0.027
	6	0.003	0.008	0.007	0.011	0.013	0.017	0.02
	7	0.019	0.018	0.023	0.019	0.021	0.023	0.024
	8	0.011	0.011	0.013	0.016	0.017	0.019	0.021
	9	0	0	0	0.002	0.007	0.013	0.018
	10	0.002	0.007	0.006	0.01	0.013	0.017	0.021

(the 2-dimensional array M formulated above) into k random samples of equal size. Out of the k samples, we use $k-1$ samples to generate a training transfer function and one sample to generate the test transfer function. The cross-validation is repeated k times where each of the 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:

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

We use $k = 2, 3, 5, 10, 20, 50, 100$ in our experiments. As the value of k increases, the number of events in the test data set reduces. For small value of $k = 2$, we divide the entire dataset in 2 sub-sets of equal size where one is used for training and the other is used for testing.

Table VII shows the MSEs for all devices in the network at various sample sizes where it can be clearly seen that the values are slightly increasing with increase in the value of k. At $k = 100$, the largest value is 0.0285, i.e., in the worst case, the error between the trained and tested values is 2.8% on average. The small error values suggest that a device behaviour (latent feature) can be captured accurately with historical PQ data from power quality meters.

IV. MICROGRID AS A 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 power quality 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 power quality class of each link where a smart meter is installed.

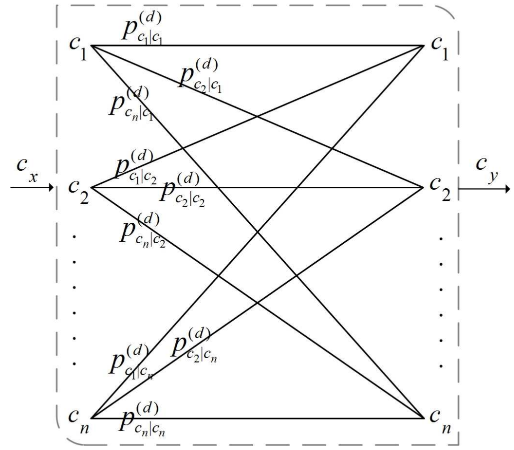


Fig. 3. Power quality transfer at each device d as a channel.

Moreover, in order to simplify our model, we treat the power flow through each node as a channel (shown in Figure 3). The input and output of this channel at each node comprises n power quality classes. The probability that a power quality 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 *power quality transfer function*, or simply *transfer function*. For a device (subnet) having multiple inputs/outputs, a power quality transfer function is associated with each input/output pair. We represent the power quality transfer 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 Bayesian network based algorithm for selecting locations for placing power meters in a power grid. The approach uses Monte Carlo sampling and probabilistic inference approaches to identify locations in the power grid which exhibit unpredictable power quality events.

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.

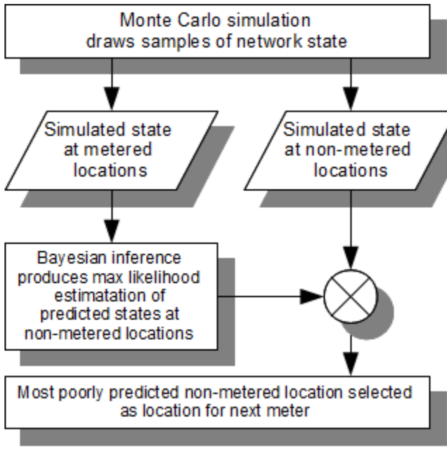


Fig. 4. Data flow diagram of meter selection process during a single iteration of the greedy algorithm.

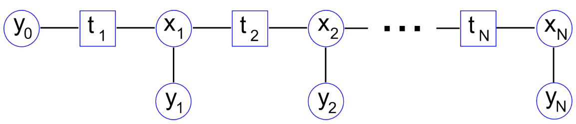


Fig. 5. Power network modeled as a factor graph.

To tackle the above challenge, we cast the problem as a Bayesian network 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 [10], since it is well understood and has been shown to work for general topologies [11] including tree networks.

1) MC Event Sampling: Given the transfer function $f(d)$, we use a Monte Carlo (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 transfer 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 $i \in \{1 \dots K\}$, we draw a sample $c_i^{(d)}$ 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 $C_i = \{c_i^{(1)}, c_i^{(2)}, \dots, c_i^{(N)}\}$ for each of the N links in the power network.

2) Event Inference using Belief Propagation: The samples obtained by the MC simulation of power quality propagation contain consistent sets of power quality values at both metered and non-metered locations. We use Bayesian inference to infer the power quality at non-metered 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 non-metered locations. This process gives a relative indication of the predictive strength on each link of the network. Figure 4 shows a high-level description of this process.

To do the prediction, we first model the power network as a factor graph (Figure 5) and then use belief propagation to find the inferred values of power quality 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 transfer 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 non-metered 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 power quality event and 0 everywhere else.

For each time slot t_i we infer the maximum likelihood power quality 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. See Algorithm 1 and Figure 4 for further details.

Algorithm 1 Monte Carlo Predicted Error Algorithm

Input: The topology T of power grid; the pmf of the input feed to first device i.e., $f_x(0)$; transfer functions $f(d)$; the number of power meters M to place; and the number of Monte Carlo samples to draw K
Output: L (list of links to be selected for meter placement)
begin
  foreach *(power meter m)* **do**
    $\epsilon_l \leftarrow 0, \forall$ links $l \in T$;
    /* ϵ_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**
          $\epsilon_l \leftarrow \epsilon_l + 1/K$; /* add $1/K$ to predicted error */
        end
      end
    end
    $selectedLink \leftarrow \max(\mathcal{E})$; /* $\mathcal{E} = \{\epsilon_l\}$ i.e., set of $\epsilon_l \ \forall \ l$ */
    L.add($selectedLink$);
  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 ψ_l';
  end
  $C_k = \{c_k^{(l)}\}$
end

B. Conditional Entropy (CE) Based Approach

Since the PQ values on network segments are dependent on each other, we exploit the idea of conditional entropy to propose another new algorithm. Further, this approach is much faster than the Bayesian network (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 re-sampling the network state after every possible meter placement.

The CE based algorithm is efficient and scalable to large scale real-world networks. Both the 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 power quality values on a link is dependent on the uncertainty of power quality values on other links (parents, childs, 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 $l_{out}^{(d)}$, if chosen for smart meter placement, influences the uncertainty of PQ values on other links, we consider the conditional entropy of all monitored links while placing power meter at a link $l_{out}^{(d)}$.

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

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

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 power quality distribution vector $f_x(d_o)$, device transition matrix $f(d_i \mid d_o)$ as:

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

where $\times$ represents the cross product, the symbol $\otimes$ represents the dot or componentwise product (also known as Hadamard product), and $\log$ is a componentwise $\log$ 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 transfer 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 transfer function $f(child \mid parent)$ is simply the product of the normal transfer functions of devices between links $l_{out}^{(d_o)}$ and $l_{out}^{(d_i)}$, i.e.,
$$F = f(\check{d_o}) \times \ldots \times f(d_i).$$
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 conditional entropy, we need to first calculate the conditional transfer function F.
We use the concept of posterior probability (the Bayes theorem) to calculate F. This function is simply the product of the reverse transfer functions of devices all the way from child to parent. The reverse transfer function $f'(d)$ (consist of $p(parent \mid child)$ or $p(X \mid Y)$) is calculated as $p(X \mid Y) = \frac{p(X)p(Y/X)}{p(Y)}$. 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:
$$f'(d) = \begin{bmatrix} f_x(\widehat{d}) \\ f_x(\widehat{d}) \\ \vdots \\ f_x(\widehat{d}) \end{bmatrix} \otimes \big[f(d)\big]^T \oslash \begin{bmatrix} f_x(d) \\ f_x(d) \\ \vdots \\ f_x(d) \end{bmatrix}^T,$$
where $\otimes$ is the componentwise product, $\oslash$ is the componentwise division, and $\widehat{d}$ is the immediate parent of device d. Finally:
$$F = f'(d_o) \times f'(\widehat{d_o}) \times \ldots \times f'(\check{d_i}).$$
3) **Devices d_o, d_i belong to different sub-trees**: 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 transfer function $f(d_r \mid d_o)$ of devices between links $l_{out}^{(d_o)}$ and $l_{out}^{(d_r)}$ using method 2. We then calculate the conditional transfer function $f(d_i \mid d_r)$ of devices between links $l_{out}^{(d_r)}$ and $l_{out}^{(d_i)}$ using method 1. Finally:
$$F = f(d_i \mid d_o) = f(d_r \mid d_o) \times f(d_i \mid d_r)$$

2) The MinEntropy Algorithm: Algorithm 2 illustrates our conditional entropy 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 $l_{out}^{(d_o)}$, we first calculate the entropy of every link $l_{out}^{(d_i)}$ given $l_{out}^{(d_o)}$. These conditional entropies are efficiently calculated by multiplying $f_x(d_o)$ with transition functions of all devices on the path between links $l_{out}^{(d_o)}$ and $l_{out}^{(d_i)}$. We do not need to

Algorithm 2 The MinEntropy Algorithm

Input: distribution function of input link to device 1 i.e., $f_x(0)$, transfer function $f(d)$, number of power meters N
Output: L (list of devices to be selected for meter placement)
begin
  foreach *(power meter m)* **do**
    $maxReduction \leftarrow 0$;
    $selectedLink \leftarrow 0$;
    foreach *(device d)* **do**
      $entReduction \leftarrow$ calcNetworkEntropy(d);
      if *maxReduction* $<$ *entReduction* **then**
        $maxReduction \leftarrow entReduction$;
        $selectedLink \leftarrow d.outputLink$;
      end
    end
    L.add($selectedLink$);
    updateEntopies($selectedLink$);
  end
end

$function$ calcNetworkEntropy(d) : $entRed$
begin
  F $\leftarrow$ identityMatrix(n);
  /* F is the combined transfer 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

explicitly identify the path from $l_{out}^{(d_o)}$ to $l_{out}^{(d_i)}$ 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 conditional entropy for a directly connecting neighbor of $l_{out}^{(d_o)}$, we then recursively calculate the entropies of neighboring links of that neighbor. Here, it should be noted that 1) 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; 2) The product of transition functions calculated from $l_{out}^{(d_o)}$ to some $l_{out}^{(d_k)}$ is used to calculate the next product; 3) 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 smart 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 are bus, switch, transformer and UPS. Power quality events are assigned a number from 1-5 in order of severity in accordance with [12] with 1 being a clean input. These are listed in Table VIII along with their descriptions. We used the transfer functions listed in Table IX and assigned a prior on the utility feed of $[\,0.9947 \quad 0.005 \quad 0.0002 \quad 0.00009 \quad 0.00001\,]$. We use two different network topologies to test our algorithms. The first is a line network of 10 (and a variant of 50) devices. The second is a tree network with 16 devices. For each topology we choose 2 different device configurations, one with all identical devices and one with a mixture of different devices. For the BP algorithm, we collect $N = 10000$ samples for each device using MC sampling. For each network configuration we place $M = 5$ meters in order of importance.

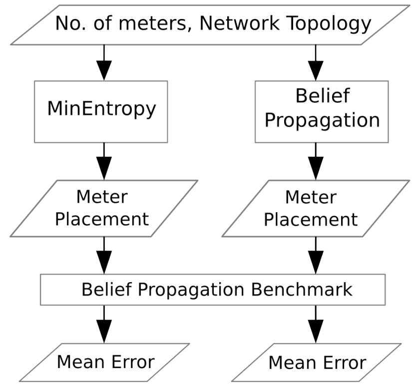


Fig. 6. An overview of the meter placement evaluation process.

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 [13]

		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

(a) Bus

		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

(b) Switch

		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

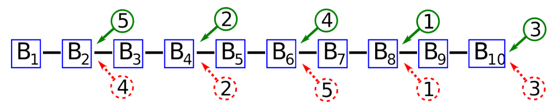
(c) Transformer

		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

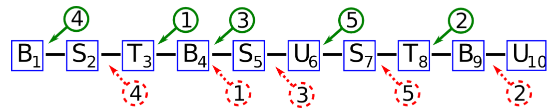
(d) UPS

TABLE X
RESULTS FOR EACH NETWORK CONFIGURATION

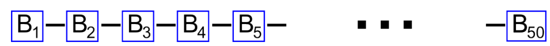
Network Topology	Algorithm	Meter Placement Sequence	Mean Error Rate	Elapsed Time (seconds)
Line homogeneous	BP	9,5,11,3,7	0.041112	270
	MinEnt	9,5,11,7,3	0.041112	0.064
Line heterogeneous	BP	5,10,6,3,8	0.040215	281
	MinEnt	4,9,5,2,7	0.064040	0.064
Long line homogeneous	BP	28,14,42,21,7	0.076345	6122
	MinEnt	28,16,40,9,21	0.076479	8.5
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



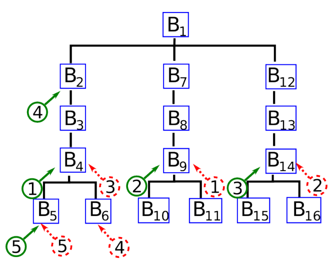
(a) Homogeous line network



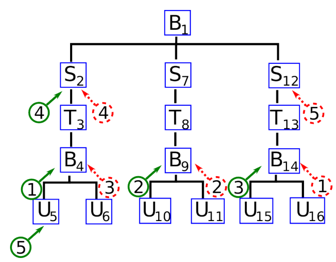
(b) Heterogeneous line network



(c) Long homogeneous line network



(d) Homogeneous tree network



(e) Heterogenous tree network

Fig. 7. Networks used in our experiments. B=bus, S=switch, T=transformer, U=UPS. Ordered dotted circles correspond with the sequence of meters placed by BP while the solid circles show the meter placed by MinEntropy.

Figure 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 (except the long 50-node network which takes about 8 seconds). On the other hand, the BP takes relatively much longer 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 non-metered locations using probability mass functions instead 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 mean-square error (MSE), i.e., the mean error between the estimated and actual transfer functions on unmonitored links. We collect a set of known samples for a given meter configuration and infer the maximum likelihood power quality event that would appear at each unmetered 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.

VII. Conclusion

Power quality 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 power quality monitoring. We formulate the problem of selecting suitable meter placements in power networks such that power quality can be best predicted. Two approaches were presented, one based on conditional entropy and one considering prediction error. Experiments in various simulation networks suggested that the conditional entropy based MinEntropy approach is much faster. Finally, the proposed solutions significantly reduce the uncertainty of power quality values on unmonitored power links.

References

[1] K. Moslehim and R. Kumar, "A reliability perspective of the smart grid," *IEEE TRANSACTIONS ON SMART GRID*, vol. 1, no. 1, june 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.

[3] Fluke Corporation, "Fluke Power Quality Analyzers," http://www.fluke.com/fluke/usen/products/categorypqttop.htm, [Online: accessed Aug. 2013].

[4] Schneider-Electric. (accessed on Aug. 2013) Smart meters. [Online]. Available: http://www.smartmeters.com/tag/schneider-electric.html

[5] IEEE, "IEEE recommended practice for monitoring electric power quality," *IEEE Std 1159-2009 (Revision of IEEE Std 1159-1995)*, pp. c1–81, June 2009.

[6] IEEE, "IEEE recommended practice for the design of reliable industrial and commercial power systems," *IEEE Std 493-2007 (Revision of IEEE Std 493-1997)*, pp. 1–383, 2007.

[7] I. T. I. C. (ITI), "ITI (CBEMA) Curve Application Note," Information Technology Industry Council (ITI), Tech. Rep., October 2007.

[8] A. Krause and C. Guestrin, "Optimizing Sensing from Water to the Web," *IEEE Computer Magazine*, August 2009.

[9] W. Yuill, A. Edwards, S. Chowdhury, and S. P. Chowdhury, "Optimal PMU placement: a comprehensive literature review," in *IEEE Power and Energy Society General Meeting*, Detroit, Michigan, July 2011.

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Pub, 1988.

[11] J. S. Yedidia, W. T. Freeman, Y. Weiss *et al.*, "Generalized belief propagation," *Advances in neural information processing systems*, pp. 689–695, 2001.

[12] EPRI, "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," *CEIDS*, 2002.

[13] C. Gamroth, K. Wu, and D. Marinakis, "A smart meter based approach to power reliability index for enterprise-level power grid," in *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE, 2012, pp. 534–539.