Outage Detection in Power Distribution Networks

Raffi Sevlian*§, Yue Zhao*‡§, Andrea Goldsmith*, Ram Rajagopal†, H. Vincent Poor‡

*Department of Electrical Engineering, Stanford University
†Department of Civil and Environmental Engineering, Stanford University
‡, ‡Department of Electrical Engineering, Princeton University
§These authors contributed equally to this work.

Abstract—An outage detection framework for power distribution networks is proposed. Given the tree structure of the distribution system, a framework is developed combining the use of real-time power flow measurements on edges of the tree with load forecasts at the nodes of the tree. Components of the network are modeled. A framework for the optimality of the detection problem is proposed relying on the maximum missed detection probability. An algorithm is proposed to solve the sensor placement problem for the general tree case. Finally, a set of case studies is considered using feeder data from the Pacific Northwest National Laboratories and Pacific Gas and Electric. We show that a 10% loss in mean detection data from the Pacific Northwest National Laboratories and Pacific tree case. Finally, a set of case studies is considered using feeder

Nomenclature

\( \mathcal{T} \) Tree network representation of a distribution feeder

\( \mathcal{T}_k \) Tree network with root node \( v_k \)

\( \overline{\mathcal{T}}_k \) Reduced subtree from decoupling non-local measurements

\((V, E)\) Vertex and edge set of tree \( \mathcal{T} \) \( (v \in V, e \in E) \)

\((V_k, E_k)\) Vertex and edge set of subtree \( \overline{\mathcal{T}}_k \)

\((\overline{V}_k, \overline{E}_k)\) Vertex and edge set of reduced sub-tree \( \overline{\mathcal{T}}_k \)

\( \mathcal{P}(v) \) Path of predecessor nodes from \( v \) to root

\( L_k \) Scalar load value at vertex \( v_k \)

\( \overline{L}_k \) Scalar forecast of load value at vertex \( v_k \)

\( \mathbf{L}, \hat{\mathbf{L}} \) Vector of loads and forecasts

\( \epsilon_k \) Forecast residual for load \( L_k \)

\( \Sigma_k \) Forecast variance for load \( L_k \)

\( \Sigma \) Forecast error covariance

\( \mathcal{M} \) Sensor placement \( (\mathcal{M} \subset E_0) \)

\( \mathcal{M}_k \) Local measurement set of reduced subtree \( \overline{\mathcal{T}}_k \) with location \( m_k \) for root of subtree.

\( \mathcal{A}_k \) Area network defined on tree with root \( m_k \)

\( h_k \) Hypothesis of outage

\( \mathcal{H} \) Set of Hypotheses of a tree network

\( \mathcal{H}(A_k) \) Set of hypotheses for area network \( A_k \)

\( \mathcal{H}(se) \) Hypothesis set associated with sensor event \( se \)

\( s \) Set of observations on edges of network

\( s_i \) \( i^{th} \) power flow measurement

\( \Delta s_k \) Effective measurement on reduced subtree \( \overline{\mathcal{T}}_k \)

\( r_{ij} \) Acceptance region of \( h_i \) over \( h_j \)

\( \mathcal{R}(h_i) \) Acceptance region of hypothesis \( h_i \) over all alternatives.

\( P_{\text{err}}(h_i) \) Missed detection probability for individual hypothesis

\( P_{\text{err}} \) Maximum missed detection probability for an area network \( A = (\mathcal{T}, \mathcal{M}) \). \( P_{\text{err}}^{\text{max}}(A) \) and \( P_{\text{err}}^{\text{max}}(\mathcal{T}, \mathcal{M}) \) used.

\( f(i, j) \) Shorthand notation for \( P_{\text{err}}^{\text{max}}(\overline{\mathcal{T}}_i, \mathcal{M}_i) \). Subtree includes vertices \( v_1, \ldots, v_j \)

\( \mathcal{SE} \) Set of sensor event, \( se \in \mathcal{SE} \)

\( W_\mu, W_\sigma \) Sum of all downstream node load mean and variances \( (\mu \text{ and } \sigma) \)

I. INTRODUCTION

Outage detection and management has been a long-standing problem in power distribution networks. As society becomes more dependent on electric power, the economic and societal costs due to loss of loads from distribution outages have been increasingly severe. Outages are mainly caused by permanent short circuit faults in the distribution system. When a short circuit fault occurs, protective devices close to the fault will automatically isolate the faulted area. The loads downstream of the protective devices will be in outage. We employ the term outage detection to denote the task of finding the status of the protective devices, and the term fault detection to denote that of finding the faults that caused the resulting outage situation.

Many methods for outage and fault detection based on artificial intelligence have been developed. Outage detection is often performed prior to fault detection and can greatly improve the accuracy of fault diagnosis. For outage detection, fuzzy set approaches have been proposed based on customer calls and human inspection [15], and based on real-time measurement with a single sensor at the substation [11]. In networks where supervisory control and data acquisition (SCADA) systems are available, a subset of the protective devices’ status can be obtained via direct monitoring. When two-way communications from the operator and the smart meters are available, AMI polling has been proposed to enhance outage detection [7]. There have also been knowledge based systems that combine different kinds of information (customer calls, SCADA, AMI polling) [6]. For fault detection, using only a single digital transient recording device at the substation, fault location and diagnosis systems have been developed based on fault distance computation using impedance information in the distribution system [16]. Using only the outage detection results, i.e., the status of the protective relays, expert systems have been applied to locate the underlying faults [4]. Incorporating voltage measurements in the distribution system with the outage detection results, fault detection methods based on knowledge based systems have been proposed [2]. Fault detection that uses during fault voltage-sag measurements and matching has been proposed in [3]. Fault diagnosis based on fuzzy systems and neural networks have also been proposed that can resolve multiple fault detection decisions [14].
Nonetheless, current practice of outage and fault detection does not provide real-time detection decisions. In addition, as the existing outage and fault detection methods based on artificial intelligence do not provide any analytical metric on how well the algorithms perform, while their performance can be evaluated numerically, it is in general hard to examine the optimality of the algorithms. Moreover, because of this lack of an analytical metric, while some of the existing approaches depend on near real-time sensing (e.g. SCADA), they do not provide guidance on where to deploy the limited sensing resources within the distribution system.

In this paper, we focus on real-time outage detection based on optimally deployed power flow sensors within the distribution system and load estimates via AMI or load forecasting mechanisms. The proposed sensing and feedback framework exploits the combination of real-time sensing and feedback from a limited number of power flow sensors and the infrequent load update from AMI or forecasting mechanisms. We develop a probabilistic model of the outage detection problem, and formulate outage detection as a hypothesis testing problem. This formulation not only allows the development of optimal detectors, but also enables an analytical metric of the overall detection performance. Based on this performance metric, we propose efficient algorithms that determine the globally optimal locations for deployment of real-time sensors, and characterize the optimal trade-off between the number of sensors to use and the optimal detection performance.

II. SYSTEM DESIGN

We consider a power distribution network that has a tree structure. The power is supplied from the feeder at the root, and is drawn by all the downstream loads. We consider an outage to a protective device isolating a faulted area. When this occurs, the loads downstream of the faulted area will be in outage. We investigate the optimal design and performance of automatic outage detection systems, with the use of the following two types of measurements:

- Load pseudo-measurements in the form of forecasts. Load forecasts will have a forecast error that must be taken into account.
- Real-time measurements of the power flows on a fraction of the lines obtained using noiseless sensors placed on the selected lines. They are modeled as noiseless sensors, since the errors of these measurements are negligible compared to those of the load pseudo-measurements.

III. SYSTEM MODEL AND NOTATION

Topology of the Distribution System: We index the buses in the distribution network by $v_1, v_2, \ldots, v_N$, with bus $v_0$ denoting the root of the tree. We index by $e_n$ the line that connects bus $v_n$ and its parent node. We denote by $\mathcal{T}_n$ the subtree with $v_n$ as the root node with $\mathcal{T}_n = (V_n, E_n)$.

Outage Hypothesis Model: Outages are modeled as disconnected edges corresponding to protective devices disconnecting loads on a network. For example we consider only single line outages in a tree with $N$ edges. In this situation, there will exist $N + 1$ hypotheses to consider: $h_i$ denotes the disconnection of edge $e_i$ while $h_0$ that denotes the non-outage situation. Let $\mathcal{H} = \{h_1, \ldots, h_N, h_0\}$ be the set of all hypothesis for a tree $\mathcal{T}$.

Load Model: Each node $v_n$ in the graph has a consumption load $L_i$. We assume for now that the load is single time invariant value. The pseudo-measurement of each load is $\hat{L}_i$. We denote the pseudo-measurement error of load $L_i$ by $\epsilon_i = L_i - \hat{L}_i$. We assume that the loads are single phase real power quantities and the errors are mutually independent random variables that follow $\epsilon_i \sim N(0, \sigma_i^2)$. Therefore, $L_i \sim N(L_i, \sigma_i^2)$. $L$ and $\hat{L}$ represent the vector of true loads and that of load pseudo-measurements; thus $L \sim N(L, \Sigma)$. The covariance matrix $\Sigma$ is considered diagonal with uncorrelated loads.

Measurement Model: For any edge $e_i$, we denote by $s$ the power flow on it to all active downstream loads. This includes the load $L_n$ associated with node $v_n$. The measured flow depends on the network topology, outage situation and the true loads. The sensor placement is denoted as $\mathcal{M}$ with $\mathcal{M} \subseteq \mathcal{E}_0$ of the full tree.

In the example in Figure [1], $\mathcal{M} = \{e_1, e_5, e_7\}$. The vector of all measurements is $s \in \mathbb{R}^{|\mathcal{M}|}$. The location of the first measurement sensor $s_1$ is $m_1 \in \mathcal{M}$, with $m_1 = \{e_1\}$.

Given a general tree $\mathcal{T}$ assume hypothesis $h_k$ corresponds to the outage of edge $e_k$ and all downstream loads. Also, assume sensor $s$ located on edge $e_m$, $(m \in \mathcal{M})$ measures the downstream load from edge $e_m$. Then the measured power consumption is given as follows:

$$s(h_k) = \sum_{e_j \in e_m \mid V_k} L_j.$$ \hspace{1cm} (1)

Area Network: Each sensor with associated downstream measurements constitutes a partition in the graph. We denote this partition as an area network. An area network $\mathcal{A} = (\mathcal{T}, \mathcal{M})$ is the tuple of a reduced subtree $\mathcal{T}$ and subtree measurements $\mathcal{M}$.

We can define the reduced graph $\mathcal{T}_i$ which represents a tree rooted at $v_i$ with the removal of all downstream subtrees rooted at a measurement.

Given the tree $\mathcal{T}_i$, which has the set of edges $E_i$, the set of all downstream sensors not including the root measurement $e_i$ is $(E_i \cap \mathcal{M}) \setminus e_i$. The union over all trees rooted at these edges will give all the downstream trees separated by a sensor. Taking the set difference will result in the reduced tree. This can be expressed as
For the example in Figure 1, an area network \( A_1 \) is constructed by taking the intersection of \( E_1 = \{e_1, \ldots, e_{10}\} \) and the set of all measurement locations \( \mathcal{M} = \{e_1, e_5, e_7\} \). By removing \( e_1 \), we have \( \{e_1, \ldots, e_{10}\} \cap \{e_1, e_5, e_7\} \setminus \{e_1\} = \{e_5, e_7\} \). Using these edges, we then remove \( T_5 \) and \( T_7 \) and finally reduced subtree \( T_1 \) is obtained.

The local measurement set \( \mathcal{M}_i \) can be constructed by the subtree \( T_i \). Let \( E_i \) and \( \mathcal{V}_i \) be the vertex and edge set of the reduced subtree \( T_i \). The set \( E_i \) can include the edges of nodes not in \( \mathcal{V}_i \). For example in Figure 1, \( E_i \) includes \( e_5 \) and \( e_7 \). We can then define the set of measurements local to the reduced subtree as \( \mathcal{M}_i = \mathcal{M} \cap E_i \). For the set \( \mathcal{M}_i = \{m_1, \ldots, m_n\} \) \( m_1 \) is the location of the root node for the area network. An area network \( A_i = (T_i, \mathcal{M}_i) \) is the tuple of reduced subtree and measurement locations at the leaves of the sub-tree.

Each area network makes an independent outage hypothesis \( \hat{h} \) independent of other area networks. Let \( \mathcal{H}(A) \) be the hypothesis associated with a particular area network. Each area network will have a hypothesis associated with an edge outage and a no-outage hypothesis.

This leads to the following \( \mathcal{H} \subset \bigcup_A \mathcal{H}(A) \cup \{b_0\} \). Each area networks set of hypotheses \( \mathcal{H}(A) \) will contain the outage of edge \( e \), but has additional hypothesis for no outage conditions in each area network.

Effective Measurement: We demonstrate in Section V that each area network requires load measurements local to the partition. In fact we show that only the difference of incoming and outgoing flows are sufficient for hypothesis detections. This quantity is defined as the effective measurement of an area. It is the sum of all the remaining active loads within \( T_k \) with flow measured at the root edge of the partition:

\[
\Delta s_k \triangleq s_k - \sum_{i:e_i \in \mathcal{M}_k \setminus m_1} s_i. \tag{3}
\]

In Eq. 3, \( s_k \) is the root measurement of network \( A_k \). We subtract from \( s_k \) the downstream measurements located in the set \( \mathcal{M}_k \setminus m_1 \).

Sensor Event: For an area network \( A \), we have a set of booleans indicating whether there is an observed power flow on a line. Boolean \( b_i \) corresponds to measurement \( m_i \in \mathcal{M} \). For each, \( b_i = 1 \) if \( s_i > 0 \) for \( m_i \) else, \( b_i = 0 \). Therefore, a single sensor event is the collection of booleans: \( \mathcal{S} = \{b_1, \ldots, b_{|\mathcal{M}|}\} \). The set of all sensor events for an area network is \( \mathcal{S}(A) \). For a given area network, clearly \( |\mathcal{S}| \leq 2^{|\mathcal{M}|} \) sensor events.

We show in Section V.C that each sensor event \( \mathcal{S} \in \mathcal{S}(A) \) in a tree network will have it’s own disjoint set of hypotheses to consider. The hypothesis set associated with a sensor event is \( \mathcal{H}(\mathcal{S}) \) with \( \bigcup_{\mathcal{S} \in \mathcal{S}(A)} \mathcal{H}(\mathcal{S}) = \mathcal{H}(A) \).

IV. GENERAL PROBLEM FORMULATION

A. Detection Problem

Given a set of power flow measurements on network edges indicated by \( \mathcal{M} \) as well as load forecasts on each node in the network, we provide a framework for detecting outages on individual edges of the tree in real time. The problem is formulated as multiple hypothesis testing problem with gaussian observations. The detector proposed is a distributed MAP detector where each area network performs its hypothesis test independent of the others.

We introduce the maximum missed detection probability over all hypotheses within an area network as \( P_{\text{err}}(A) = \max_{h_k \in \mathcal{H}(A)} \Pr(\hat{h} \neq h_k | h_k \text{ is true}) \). However, our global constraint is the network missed detection probability

\[
P_{\text{err}}(\mathcal{T}, \mathcal{M}) = \max_A \max_{h_k \in \mathcal{H}(A)} \Pr(\hat{h} \neq h_k | h_k \text{ is true}). \tag{4}
\]

Throughout this work we use \( P_{\text{err}}(A) \) and \( P_{\text{err}}(\mathcal{T}, \mathcal{M}) \) interchangeably for convenience.

B. Sensor Placement Problem

For any given probability target \( P_{\text{target}} \), we find the minimum number of sensors that keep the maximum missed detection probability below \( P_{\text{target}} \). This is formulated as:

\[
\min M \tag{5a}
\]

s.t. \( \exists \mathcal{M}, |\mathcal{M}| \leq M \tag{5b}
\]

\[
P_{\text{err}}(\mathcal{T}, \mathcal{M}) \leq P_{\text{target}} \tag{5c}
\]

The optimization (5) is a combinatorial optimization over \( \mathcal{M} \) which is in general hard to solve. Enumerating all potential \( \mathcal{M} \) instances yields \( (N)^M \) possible sensor placements which for large \( N \) is undesirable. Typically, values of \( N \) can be quite large, so efficiently placement methods are desired. In section VIII-A we show that optimization (5) can be solved efficiently.

V. PROPERTIES OF THE DISTRIBUTED MAP DETECTOR

Fig. 2. Measurement decoupling shown in the case of a [a] line network and [b] tree network. (All vertices of the network are not shown.)

A crucial property of the proposed outage detection scheme is that each area network can independently perform hypothesis detection and we are not constrained to single outage cases. This leads to two very important properties which simplify the detection and placement problems.

Lemma 1: (3) Given a vector of observations \( s_i|_{h_i} \sim N(\mu_i, \Sigma_i) \) for all \( i \) in some set. To distinguish between any two hypotheses \( h_i \) and \( h_j \) we compute the log likelihood ratio \( LLR_{ij}(s) \) with the detector performing the following:
\[
\hat{h} = \begin{cases} 
  h_i, & \text{LLR}_{ij}(s) \geq \eta_{ij} \\
  h_j, & \text{otherwise}
\end{cases}
\]

(6)

\[
\text{LLR}_{ij}(s) = (s - \mu_j)\Sigma_i^{-1}(s - \mu_j) - (s - \mu_j)\Sigma_j^{-1}(s - \mu_j) - \frac{1}{2} \log \left( \frac{|\Sigma_j|}{|\Sigma_i|} \right)
\]

(7)

In the case of equal priors this leads to \( \eta_{ij} = 0 \). Therefore for a single pairwise test, \( h = h_i \) if \( s \in r_{i,j} \), where \( r_{i,j}(s) = \{ s \in R^{|M|} : \text{LLR}_{ij}(s) > \eta_{ij} \} \). This notation can be extended to the multiple hypothesis case.

**Corollary 1:** \( \forall h_i \in \mathcal{H}(A) \) the acceptance region for the area network MAP detector is \( \mathcal{R}(h_i) = \{ s \in R^{|M|} : \forall h_j \in \mathcal{H}(A) s \in r_{i,j} \} \).

The acceptance region for hypothesis \( h_i \) is equivalent to the intersection of each pairwise region:

\[
\mathcal{R}(h_i) = \bigcap_{j \in \mathcal{H}(A)} r_{i,j}
\]

(8)

Notice this is the same as saying the likelihood of \( h_i \) is greater than all other alternatives. The \( s \) observation space will always have a unique maximizing \( h_i \) due to the inequality in Eq. (7).

**A. Decoupling in a Line Network**

First consider the line network in Figure 2(a) with two sensors \( s_1 \) and \( s_2 \) in locations \( m_1 \) and \( m_2 \) respectively. Consider only area network \( A_2 \), with root edge \( m_2 \) where it must perform outage detection on hypothesis \( \forall h_1, h_j \in \mathcal{H}(A_2) \). Given the complete set of measurements \( s = (s_1, s_2) \) we can show that \( s_2 | h_1 \sim N(\mu_i, \sigma_i^2) \) where

\[
\mu_i = \sum_{v_j \in V_{m_2} \setminus V_i} \hat{L}_{ij}, \quad \sigma_i^2 = \sum_{v_j \in V_{m_2} \setminus V_i} \sigma_j^2.
\]

(9)

Also, \( s_1 = s_2 + N(\mu_e, \sigma_e) \) where

\[
\mu_e = \sum_{v_i \in V_{m_1} \setminus V_{m_2}} \hat{L}_i, \quad \sigma_e^2 = \sum_{v_i \in V_{m_1} \setminus V_{m_2}} \sigma_i^2.
\]

(10)

This leads to the following joint distribution for the complete observation set.

\[
\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} | h_1 \sim N \left( \begin{bmatrix} \mu_i + \mu_e \\ \mu_i \end{bmatrix}, \begin{bmatrix} \sigma_i^2 + \sigma_e^2 & \sigma_i^2 \\ \sigma_i^2 & \sigma_e^2 \end{bmatrix} \right)
\]

(13)

Under this explicit model, we can compute the log likelihood ratio in eq. (7), which after considerable simplification is independent of \( s_1 \). It is in fact identical to the LLR under the following hypothesis test: \( h_1: s \sim N(\mu_i, \sigma_i) \) and \( h_2: s \sim N(\mu_j, \sigma_j) \).

We can extend this to all upstream measurements beyond the first. An intuitive argument is that we make two observations, one of the signal of interest and of the same signal value with additive noise. Therefore, we can do no worse by discarding the noisy signal.

**B. Decoupling in a Tree Network**

Next we consider the tree network in Figure 2(b). We can construct the following observation set similar to the line network:

\[
\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} N(\mu_i, \sigma_i^2) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} N(\mu_e, \sigma_e^2)
\]

(14)

Which leads to the following joint distribution:

\[
\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} | h_1 \sim N \left( \begin{bmatrix} \mu_i + \mu_e \\ \mu_i + \mu_e \end{bmatrix}, \begin{bmatrix} \sigma_i^2 + \sigma_e^2 & \sigma_i^2 \\ \sigma_i^2 & \sigma_e^2 \end{bmatrix} \right)
\]

(15)

Again, evaluating the log likelihood ratio e.g. (7), leads to a expression which depends only on \( \Delta s = s_1 - s_2 \). Therefore the effective measurement for the network is the sufficient statistic for the MAP detector.

**C. Hypothesis Set Reduction**

In constructing an acceptance region for \( h_i \) we can omit hypothesis all outage \( h_k \) upstream from \( s_2 \). For example consider the computation of \( \text{LLR}_{L_{ij}}(s_1, s_2) \).

When \( h_i \) and \( h_j \) (or any other hypothesis on the same branch as \( h_i \)) are being compared, the joint distribution \( (s_1, s_2)^T \) is non-degenerate. Therefore, there is a likelihood of confusion between the two hypothesis. However the joint distribution under \( h_k \) is degenerate since \( s_1 = 0 \) regardless of \( s_1 \). Therefore, we can distinguish the two alternatives without error. Computational details of hypothesis reduction for a tree network with arbitrary sensor positions are discussed in Section IX-A1.

**VI. ILLUSTRATIVE EXAMPLE**

![Diagram for a distribution network with a tree structure. The following examples shows a 10 node network (V_0 not included). The sensor placement is M = {e_1, e_4, e_10}.](image)

Consider the following 10 node line network in Figure 3. First we assume all loads in the network are assumed known with \( L_i = 1, \forall i \). Also assume a fixed sensor placement for now with \( M = \{e_1, e_4, e_10\} \).

We consider only single line outages. Assuming a single outage condition and perfect load knowledge, we can enumerate for each area network the possible outage conditions as well as the non-outage conditions.

Table II illustrates such a procedure which is just a consequence of conservation of energy. For example, the relation for \( s_1 \) can be evaluated by eq. (16) or explicitly by \( s_1(h_k) = \sum_{v_j \in V_{m_1} \setminus V_{m_2}} L_j \).

We also compute the effective measurements for each area network: \( \Delta s_1, \Delta s_2 \) and \( \Delta s_3 \). Therefore, if we know each load exactly as well as the edge flows. We can determine every possible outage in the network with no uncertainty from a table lookup.

A lookup table is unfortunately untenable, since we do not know every load value precisely. Substituting known load values with forecasts of loads allows us to tradeoff uncertainty in outage detection with removing the burden of measuring in realtime every node in the network. So given, \( L \sim N(L, \Sigma) \) we can construct a probability distribution describing how likely an observation of flow should be conditional on the forecasted values.
This leads to the following observation about the probability distribution function for each sensor measurement:

\[ \Delta s_n(h_k) \sim N \left( \sum_{j \in \Delta C_k^n} \hat{L}_{ij}, \sum_{j \in \Delta C_k^n} \sigma_j^2 \right) \]  \hspace{1cm} (16)

Here \( \Delta C_k^n \) is the following subset of \( V \) obtained by evaluating Eq. (3) for each area network.

Therefore, given the forecast values and uncertainty of the forecast, we have a probabilistic description of what the realtime flow measurement should be. Table III illustrates such a procedure. Here, we assume each load \( \hat{I}_i = 1 \) and each forecast standard deviation is \( \sigma_i = 0.1 \). With abuse of notation, \((\mu, \sigma^2)\) denotes a gaussian random variable with mean, \( \mu \) and variance \( \sigma^2 \).

### Table II

| \( h_k \) | \( \mu_{\Delta s_1}, \sigma_{\Delta s_1} \) | \( \mu_{\Delta s_2}, \sigma_{\Delta s_2} \) | \( \mu_{\Delta s_3}, \sigma_{\Delta s_3} \) | \( P_{err}(h_k) \) |
|---|---|---|---|---|
| \( A_1 \) | \( e_1 \) | \((0,0)\) | \((0,0)\) | \((0,0)\) | 0 |
| | \( e_2 \) | \((1,0)\) | \((0,0)\) | \((0,0)\) | 0.20 |
| | \( e_3 \) | \((2,0)\) | \((0,0)\) | \((0,0)\) | 0.40 |
| | \( \emptyset \) | \((3,0)\) | \((0,0)\) | \((0,0)\) | 0 |
| \( A_2 \) | \( e_4 \) | \((3,0)\) | \((0,0)\) | \((0,0)\) | 0 |
| | \( e_5 \) | \((3,0)\) | \((1,0)\) | \((0,0)\) | 0.20 |
| | \( e_6 \) | \((3,0)\) | \((2,0)\) | \((0,0)\) | 0.70 |
| | \( e_7 \) | \((3,0)\) | \((3,0)\) | \((0,0)\) | 0.77 |
| | \( e_8 \) | \((3,0)\) | \((4,0)\) | \((0,0)\) | 0.80 |
| | \( e_9 \) | \((3,0)\) | \((5,0)\) | \((0,0)\) | 0.49 |
| | \( \emptyset \) | \((3,0)\) | \((6,0)\) | \((1,0)\) | 0 |
| \( A_3 \) | \( e_{10} \) | \((3,0)\) | \((0,0)\) | \((0,0)\) | 0 |
| | \( \emptyset \) | \((3,0)\) | \((1,0)\) | \((0,0)\) | 0 |

Table III shows the local observations in each local area. For each area the appropriate \( \Delta s_n(h_k) \) is a gaussian random variable. Within each area network, the mean missed detection probability is used to construct the MAP detector.

Figure 4 illustrates the MAP acceptance regions \( R(h_k) \) for each potential hypothesis. For each of the hypotheses of an area network, we can compute missed detection probability \( P_{err}(h_k) = \text{Pr}(\hat{h} \neq h_k | h_k \text{ is true}) \forall h_k \in H(A). \)

Table II shows the computed missed detection probabilities for each hypothesis in an area network. For example consider area network \( A_1 \). Notice that although the observation distributions for \( h_k = e_3 \) and \( h_k = \emptyset \) show overlapping likelihood functions. However, there will be no error between the two. They are perfectly distinguishable. This is because area network \( A_1 \) has access to \( s_1 \) and \( s_2 \). So in the case of \( h_k = \emptyset \), there is positive power flow through \( s_2 \) while if \( h_k = e_3 \), we will be noticed in sensor \( s_2 \). Hence, we can distinguish these end conditions. This is the importance of specifying the sensor events which lead to perfect hypothesis detection.

### VII. Properties of Maximum Missed Detection

Here we can show the function \( P_{err}^{\text{max}}(T, \mathcal{M}) \) used in optimization 5 is separable and non-decreasing. These properties are used in providing optimality guarantees of our proposed algorithm for both the line and tree network.

**Proposition 1:** Given a network \( T \) and placement \( \mathcal{M} \) we can decompose the maximum missed detection error

\[ P_{err}^{\text{max}}(T, \mathcal{M}) = \max(P_{err}^{\text{max}}(T, \mathcal{M}_1), \ldots, P_{err}^{\text{max}}(T, \mathcal{M}_M)) \]  \hspace{1cm} (17)

This is a consequence of the decentralized detector used in our problem formulation. The constraint function in eq. 4 takes the maximum of each area. Each area makes an independent decision, and only considers hypotheses within an area.

Next consider the increasing nature of the \( P_{err}^{\text{max}}(T, \mathcal{M}) \)
**Definition 1:** Two area networks are nested $\mathcal{A} \subset \mathcal{A}'$ for a given underlying network $\mathcal{T}$ if for each reduced subtree $\mathcal{T}'$ we have $\hat{V} \subset \hat{V}'$.

**Lemma 2:** Given two area networks $\mathcal{A}$ and $\mathcal{A}'$ where $\mathcal{A} \subset \mathcal{A}'$, $P_{\text{err}}^{\text{max}}(\mathcal{A}) \leq P_{\text{err}}^{\text{max}}(\mathcal{A}')$.

The nested area definition allows us to properly define the non-decreasing nature of the maximum missed detection probability.

**Proof:** Given $h_i$ is the hypothesis with the largest error in $\mathcal{A}$ and $\mathcal{A}'$. The missed detection error in $\mathcal{A}$ is $P_{\text{err}}(h_i) = 1 - \text{Pr}(\cap_{h_j \in H(\mathcal{A})} \{\forall e \in r_{i,j} \}|h_i|$ is true). So for the $\mathcal{A}'$ we have:

$$1 - P_{\ell}(h_i) = \text{Pr}(\cap_{h_j \in H(\mathcal{A}') \setminus H(\mathcal{A})} \{\forall e \in r_{i,j} \}|h_i$ is true)

$$= \text{Pr}(\cap_{h_j \in H(\mathcal{A}') \setminus H(\mathcal{A})} \{\forall e \in r_{i,j} \}|h_i$ is true)

$$\leq \text{Pr}(\cap_{h_j \in H(\mathcal{A})} \{\forall e \in r_{i,j} \}|h_i$ is true)

$$= 1 - P_{\text{err}}(h_i).$$

Therefore, $P_{\text{err}}(h_i) \leq P_{\text{err}}^{\text{max}}(h_i)$. If another hypothesis is greater than $h_i$ in $\mathcal{A}'$, then the inequality obviously still holds.

Intuitively, the larger the area, the more alternatives we must consider. Therefore, the maximum error will will always have a smaller acceptance region in the larger network.

For the line network case, we introduce a shorthand and is used in the description of the line algorithm as well as the proof due to simplicity of exposition. Given an area network for a line, where $\mathcal{A} = \{\mathcal{T}_i, M_i\}$ and $M_i = \{e_i, e_j\}$. This is a line network that is characterized by a root node $v_i$ and terminal node $v_j$. We introduce $f(i, j) = P_{\text{err}}^{\text{max}}(\mathcal{A})$ for the this line network.

Therefore the separable property in $1$ equivalent to $f(i, j) = \max(f(i, p), f(p + 1, j))$ if a sensor is placed in location $p$. Likewise, lemma $2$ is equivalent to $f(i, j) = f(i, j + 1)$ and $f(i, j) \leq f(i - 1, j)$.

### VIII. LINE ALGORITHM AND GENERAL ALGORITHM

**FRAMEWORK**

For the line network we present an algorithm for solving optimization $\mathcal{A}^a$ and use it to show a procedure for a greedy algorithm in the general tree case.

1) **Algorithm Description:** The intuition of the algorithm is the following:

The sub function $f(i, j)$ on line $\mathcal{A}$ will do the following. For each subtree problem $f(i, j)$, enumerate each potential hypothesis $\mathcal{H}(\mathcal{A})$. For each hypothesis $h_k$ compute the MAP decision boundary $\mathcal{R}(h_k)$ and evaluate the missed-detection error: $P_{\text{err}}(h_k) = \text{Pr}(\Delta \not\in \mathcal{R}(h_k)|h_k$ is true). Test maximum error $\max_{h_k} P_{\text{err}}(h_k)$ against feasibility condition $P_{\text{target}}$.

Starting at the edge $e_N$ farthest from the root node, the algorithm considers a potential area network $f(e_c, M_{\text{last}})$. Given the set of current measurements, $M_{\text{last}}$ is the most recent measurement placed. It is also the terminal measurement for the current subtree. If the maximum error of the network is less than $P_{\text{target}}$ we move onto the next root node. Else, we place a sensor and update $M$.

**Theorem 1:** Algorithm $\mathcal{A}$ returns a set of sensor locations that achieves $P_{\text{target}}$ with the minimum number of sensors, and hence solves optimization $\mathcal{A}^a$.

**Proof:** See Appendix $\mathcal{A}$.

### IX. PLACEMENT ALGORITHM IN TREE NETWORK

The framework described in Section VIII-B can be implemented efficiently with Algorithm $\mathcal{A}$. The inputs to the method are the tree network $\mathcal{T}$ and the set of nominal load forecasts $\hat{L}$ and forecast variance $\Sigma$.

To have the effect of starting at the leaf of the network and move our way up to the root, the algorithm will process a set of nodes $\mathcal{V}_{\text{process}}$. For example in in Figure $\mathcal{V}_{\text{process}} = \{v_8, v_9, v_{10}, v_6, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_1, v_2, v_3\}$.

We generate $\mathcal{V}_{\text{process}}$ in line $\mathcal{A}$ with generate-node-order. The function takes the tree $\mathcal{T}$ and traverses via breadth first search keeping track of the depth of each node. Reversing this list of depths yields a list of nodes to process.

In line $\mathcal{A}$ we iterate over the current root node $v_r$ and current sensor placement $\mathcal{M}$. In line $\mathcal{A}$ we construct the current area network $\mathcal{A}$ with the reduced subtree $\mathcal{T}$ and local measurement $\mathcal{M}$. We then evaluate $P_{\text{err}}^{\text{max}}(\mathcal{A})$ in line $\mathcal{A}$, the details of the function evaluation are explained in Section IX-A.
Algorithm 2: Solution to optimization \(5a\) for tree network.

Result: Optimal Placement for general tree network

Input: [1] Tree network \(T\) [2] Nominal loads statistics \(L, \Sigma\) [3] Subproblem Ordering \(V_{\text{process}}\)

1) \(\star\) Generate node process ordering \(\star\)
2) \(V_{\text{process}} \leftarrow \text{generate-node-order}(T)\)
3) \(\star\) Initialize sensor placement as empty \(\star\)
4) \(M \leftarrow \emptyset\)

for \(v_r \in V_{\text{process}}\) do
   /* Construct Area Network with root at \(v_r\) */
   \(A \leftarrow (T_{v_r}, M)\)
   /* Evaluate the current subtree maximum missed */
   \(s \leftarrow \text{line-action}(A)\)
   if \(P_{\text{err}}^{\max}(s) \leq P_{\text{target}}\) then
      /* continue to next node */
      \(\star\)
   else if \(|\text{child}(v_r)| = 1\) then
      \(M \leftarrow \text{line-action}(A)\)
   else if \(|\text{child}(v_r)| > 1\) then
      \(M \leftarrow \text{tree-action}(A)\)
   end
end

Fig. 5. Flowchart describing methodology of generating missed detection probabilities for general subtree. Three steps involved include constructing sensor-events, \((\mu_k, \sigma_k^2)\) pairs and final error probability calculations.

If \(P_{\text{err}}^{\max}(A) > P_{\text{target}}\) we perform a placement action \(\text{line-action}\) or \(\text{tree-action}\) depending on the number of child nodes of \(v_r\). The routines \(\text{line-action}\) and \(\text{tree-action}\) determine the location of the next sensor placement and are described in Section IX-B.

A. Evaluating \(P_{\text{err}}^{\max}(A)\) for tree networks

The details for evaluating the maximum missed detection probability for a general area network have been omitted so far. Evaluating \(P_{\text{err}}^{\max}(A)\) follows the general procedure in Figure 5

Given \(A = (T, M)\) with terminal flow measurements, generating the full vector of probabilities involves three steps. (1) Enumerating the set of sensor events \(SE = \{s_1, \ldots, s_n\}\) and performing hypothesis set reduction for each \(s\) to construct \(H(s)\) (Section IX-A1). For each \(H(s)\) generate the mean variance pairs \((\mu_k, \sigma_k^2)_{h_k \in H(s)}\). (Section IX-A2). Finally, we compute the acceptance region for each hypothesis \(R(h_i)\) and missed detection probabilities for each hypothesis \(P_{\text{err}}(h_i)\) returning only the largest.

1) Hypothesis Set Reduction on Subtree: In Section IX-C we show that a MAP detector using hypothesis set reduction is still optimal. Here we present a computational method of evaluating each set of hypothesis corresponding to unique sensor events in an area network.

We use the following graph structure.

Definition 2: Let \(P(v_i)\) be the unique path from the root of the subtree to node \(v_i\).

Then the following holds.

Proposition 2: Given a set of sensor measurements \(s = \{s_1, \ldots, s_n\}\) in locations \(M = \{m_1, \ldots, m_n\}\) then \(v_i \in H(se)\) if \(v_i \in P(m)\) for all \(m > 0\) and \(v_i \notin P(s_m)\) for all \(m = 0\).

Here we abuse notation by stating \(v_i \in H(se)\).

Therefore for a given \(se\) which contains the set of positive and zero flows, we include outage hypotheses only if they (1) lie in the path of a zero flow measurement and (2) don’t lie in the path of a positive flow measurement.

This leads to the following naive method of generating the set \(SE\) by evaluating the following expression:

\[
V \bigcap \left( \bigcap_{s_i = 0} P(s_i) \right) \setminus \bigcup_{s_i > 0} P(s_i) \quad (24)
\]

for each non-trivial sensor. We must evaluate each sensor event with a positive root flow and then evaluate eq. (24) which leads to \(2^{|M| - 1} - 1\) possible events. The final case with \(s_1 = 0\) is not considered since it’s the trivial \(se = \{0 \ 0 \ 0\}\).

Fig. 6. Hypothesis set reduction procedure performed on tree network with one root and two terminal sensors. The vector \(se = \{b_1 \ b_2 \ b_3\}\) indicates the flows on each measurement. Each flow can be either positive or zero. Subfigures show the following: (a) \(se = \{1 \ 0 \ 0\}\), (b) \(se = \{1 \ 0 \ 1\}\), (c) \(se = \{1 \ 1 \ 0\}\), (d) \(se = \{1 \ 1 \ 1\}\). Highlighted portion of the graph indicate possible hypothesis to consider for the sensor event.

Eq. (24) is cumbersome but can be interpreted easily. The set \(V_m\) is all possible outages within a subtree. Also, only elements in \((\bigcap_{b_i = 0} P(s_i))\) can be included since the single outage was sensed by some sensors observing zero flow. Therefore we must take the
common node outages. Finally, any paths leading to the positive, \((\cup_{h_{k-1}} P(s_{i}))\), value cannot be included since if they were, a zero flow would have been recorded. Therefore they must all be removed. So we take a union of all the hypothesis and finally take a set difference.

An example tree in Figure 6 shows a reduced tree \(\mathcal{T}\) with placement \(\mathcal{M}\) with root sensor \(s_1\) and terminal sensors \(s_2\) and \(s_3\). Each sensor can measure a definite positive flow or none at all. Consider Table IX-A1 shows the enumeration of each binary sensor-event may lead to a combinatorially intractable calculation. Also, almost all of these of these events lead to an empty set of hypothesis.

### Algorithm 3: Hypothesis Set Reduction for Tree Network.

**Result:** Compute Set of Sensor Events \(\mathcal{SE}\)

**Data:** [1] \(\mathcal{V}\) subtree nodes

[2] \(\mathcal{S}\) Terminal sensor of subtree

[3] \(P(s)\) terminal sensor path list

1. /* initialize structures */
2. \(\mathcal{SE} \leftarrow \{\emptyset\}\)
3. \(\mathcal{H} \leftarrow \{\emptyset\}\)
4. for \(v \in \mathcal{V}\) do
5. \(k \leftarrow 0\)
6. for \(m \in \mathcal{M}\) do
7. if \(v \in P(s)\) then
8. \(b_k \leftarrow 1\)
9. else
10. \(b_k \leftarrow 0\)
11. end
12. end
13. \(k \leftarrow k + 1\)
14. /* construct actual sensor event */
15. \(se \leftarrow \{b_1, \ldots, b_{|\mathcal{M}|}\}\)
16. if \(se \notin \mathcal{SE}\) then
17. /* if se not in set SE add it */
18. \(\mathcal{SE} \leftarrow \mathcal{SE} \cup se\)
19. end
20. /* finally, add outage hypothesis */
21. \(\mathcal{H}(se) \leftarrow \mathcal{H}(se) \cup h_v\)
22. end

Algorithm 3 provides a method of evaluating the set \(\mathcal{SE}\) and \(\mathcal{H}(se) \forall se \in \mathcal{SE}\) in an \(O(\mathcal{M}N^2)\) complexity. Starting in line 4 the algorithm will look at every potential node (or edge outage). For each node, it will evaluate the possible sensor event that will be observed if that edge is in outage (line 5). Finally, if not already added to the output, we will output the \(se\). Else just add the hypothesis associated with the \(se\).

2) **Subtree Mean-Variance pairs:** Given the initial tree \(\mathcal{T}\) with vertex weights given as the nominal load forecast \(\mathbf{L}\) and covariance \(\Sigma\), we can compute a function \(W(n)\) which gives the total downstream load from each node in the network. Specifically \(W_{\mu}(n) = \sum_{k \in V_i} \bar{L}_k\) and \(W_{\sigma^2}(n) = \sum_{k \in V_i} \sigma_k^2\) which be constructed by a \(O(N)\) traversal of the tree. Previously we explicitly stated the sets to sum over, however for computational purposes, precomputing these graphs is beneficial.

We can compute the values \(\Delta s(h_i) \sim N(\mu(h_k), \sigma^2(h_k))\) for the area network where

\[
\mu(h_k) = W_{\mu}(m) - \sum_{m \in \mathcal{M} | m_r} W_{\mu}(h_k) \quad (25)
\]

\[
\sigma^2(h_k) = \mu^2(h_k) - \mu^2(h_k) \quad (26)
\]

The values for \(\sigma^2(h_k)\) are computed in an identical manner using \(W_{\sigma^2}\) instead.

3) **Evaluating Subtree Probabilities:** We now have a set of \(\mathcal{H}(se) \forall se \in \mathcal{SE}\) and \(\{\mu, \sigma^2\}\) \(h_k \in \mathcal{H}(se)\). For a given \(h \in \mathcal{H}(se)\) we compute \(\mathcal{R}(h_k)\) using only the set of alternative hypotheses in \(\mathcal{H}(se)\). The naive method involves including all alternatives which leads to a \(O(N^2)\) procedure. An efficient method will compute \(\mathcal{R}(h_k)\) by considering only alternative hypothesis \(h_j\) where \(\mu_j \in [\mu_k - 3\sigma_k, \mu_k + 3\sigma_k]\).

### B. Sensor Placement Action

Implementing the framework for the greedy upstream traversal using the \(V_{\text{process}}\) structure leads to different placement actions for lines and trees.

![Correct node traversal in the line network starts from the leaf and moves up to the root. Therefore evaluate \(A\) then move up to \(A'\). This can be performed in line-action by placing a sensor at the child of the violating node.](image)

1) **line-action:** Given that our current subproblem satisfies \(P_{\text{err}}(\mathcal{T_i}, \mathcal{M}) < p_{\text{target}}\), If the next subproblem does not, our only option is place a sensor at \(\text{child}(v_i)\). Notice this is exactly what is happening in Algorithm 1.

2) **tree-action:** Given that the algorithm up to now has placed sensors on the two disjoint trees with roots with \(v_1\) and \(v_2\). Then we must move as far up to the root as possible before we are forced to place a measurement.

This leads to two different types of actions. The first, we call **myopic-tree-action** which is suboptimal but leads to a bounded combinatorial method. The second is **optimal-tree-action** which is the optimal method of the greedy algorithm, but unfortunately has exponential complexity in the worst case.

3) **myopic-tree-action:** Figure 8 shows an example of a tree network. Assuming that the individual children have already been processed: \(P_{\text{err}}(A_1) < p_{\text{target}}\) and \(P_{\text{err}}(A_2) < p_{\text{target}}\). We want to construct an area network rooted at \(v_1\) with the least number of sensors.

Assuming that \(P_{\text{err}}(A_1) > p_{\text{target}}\), we have the choice of placing a sensor at either \(v_1\) or \(v_2\). We enumerate each choice \(A_1', A_2\) and choose the one with the smallest \(P_{\text{err}}\).
4) **Sub optimality of myopic-tree-action:** To correctly solve the subtree optimization the aim was to get as close to root node with as few sensors as possible. It turns out unfortunately that the myopic strategy is suboptimal.

![Diagram](image-url)

**Fig. 8.** Correct node traversal in the tree network assumes $P_{\text{err}}(A_0) < p_{\text{target}}$ and $P_{\text{err}}(A_1) < p_{\text{target}}$. If $P_{\text{err}}(A_1) > p_{\text{target}}$ we do nothing. Else, we must choose one of the $2^{\text{child}(v_r)} - 1$ area networks that can be constructed for example $A'_1$ and $A'_2$.

However it turns out that choosing $A_1$ was the optimal choice since under that choice the test $P_{\text{err}}(A_1) < p_{\text{target}}$ so the greedy of choosing a placement with the minimum probability of error does not guarantee that we can traverse farthest compared to other choices.

5) **optimal-tree-action**: Given the sub optimality of the myopic method, the optimal greedy method has to enumerate each intersection state space and maintain all $2^{\text{child}(v_r)} - 2$ possible trees. Then each subtree will be processed until a sensor is placed. Finally when all the subtrees are processed the subtree closest to the root is chosen as the candidate. All other subtrees are pruned.

**X. EXPERIMENTAL DISCUSSION**

We present the following set of analysis:

1) Present a set of statistical models for distribution loads and load forecasts. This relies on previously published work as well as analysis from PG & E smart meter data from northern California.

2) General sensitivity analysis for a randomly generated network. This gives general insights on sensitivity of model parameters for line and tree networks. Analysis of mean errors is also performed.

3) Placement on the PNNL sample distribution feeders.

**A. Modeling Assumptions**

In [13], the authors present a set of modeling assumptions used in this work. They model the distribution of loads which can be disconnected by use of switches and fuses in a typical distribution feeder. Also, they present a model showing the typical forecast accuracy (measured in coefficient of variation) for loads in the distribution network. This accuracy follows a scaling law which has been presented in other work as well [11] [12]. The analysis is verified using feeder and load data from Pacific Northwest National Lab. Forecast scaling is validating in PG & E data we rely on the following set of assumptions in the simulation study.

1) Loads on the distribution system follow a generalized Pareto distribution:

$$f(x; \kappa, \sigma, \theta) = \left( \frac{1}{\sigma} \right) \left( 1 + \frac{x - \theta}{\sigma} \right)^{-(\kappa-1)/\kappa}$$

(27)

2) Forecast coefficient of variation for a day ahead forecaster is dependent on the mean load of the group following:

$$\bar{\text{CV}}(W) = E[\text{CV}(x_A, \hat{x}_A) | W_A = W]$$

$$\bar{\text{CV}}(W) = \sqrt{\frac{\beta_0}{W} + \beta_1} \%.$$  

(28)

Reasonable fit is given as

$$\bar{\text{CV}}(W) = \sqrt{\frac{3562}{W} + 4.9}.$$  

(29)

Or the general rule of thumb:

$$\text{CV} \approx \max \left\{ \sqrt{\frac{50}{\text{Residential homes}}}, 6.5\% \right\}$$

3) Forecast residuals are normally distributed and uncorrelated in time.
B. Sensor Placement Sensitivity Analysis

We simulate the placement method on a randomly generated tree to show the sensitivity of various key parameters. The analysis computes the sensor density (SD) which measures $\#$ sensors per total nodes. From optimization we the model parameters needed in the optimization are the following:

- graph topology $T$
- mean load forecast of each node, $\hat{L}$
- forecast error for each node $\Sigma$

We look to evaluate $SD(T_N, \hat{L}, \Sigma, P^{target})$. It is difficult to evaluate this expression for sensitivity since $\hat{L}$ and $\Sigma$ takes a multidimensional input. Since $\Sigma$ can be modeled as $\Sigma = diag(\kappa L)$ a one dimensional sensitivity parameter, $\kappa$ is available. The value of $\kappa$ is equivalent to the coefficient of variation for a forecaster. We keep this as a free parameter here for a general analysis. To simulate the spread of loads we model each load to follow $L_i \sim f(L; \kappa, \sigma, \theta)$.

Given this parameterization, we evaluate $SD(T_N, \tau, \kappa, P^{target})$. These results are repeated over $MC$ instantiations. Multiple runs are performed to generate smooth curves for the underlying system parameters. Therefore given a value of $\tau$, $\kappa$, and $P^{target}$, we perform the following MC times.

Algorithm 4: Sensitivity Analysis Procedure

Result: Compute Sensor Events

Input: [1] Tree Network $T_N$
[2] Error Target: $P^{target}$
[3] Coefficient of Variation: $\kappa$
[4] Mean Load Variance: $\tau$

Output: $SD(T_N, \hat{L}_{mc}, \Sigma_{mc}, P^{target})$

1) Generate loads which are normally distributed $L_{i,mc} \sim f(L; \kappa, \sigma, \theta)$.
2) Given a set of load values, set the forecast uncertainty $\Sigma_{mc} = diag(\kappa L_{mc})$.
3) Set appropriate error target: $P^{target}$
4) Solve optimization (5) via Algorithm 4

The final reported sensor density value is:

$$SD(\tau, \kappa, P^{target}) = \frac{1}{MC} \sum_{mc=1}^{MC} SD(T_N, \hat{L}_{mc}, \Sigma_{mc}, P^{target})$$ (30)

with graph subscript suppressed.

Forecast $\kappa$ is an important metric to consider since it shows the effect of increasingly accurate forecasting in the detection process. If loads in each node are known exactly, a single measurement is needed to detect any change in the tree. However this is not the case, so the analysis here looks at values ranging from 1% to 30%. Given the aggregation error curve, $30\%$ relative error is typical at the home level. The irreducible error in $SD$ is 5%, however it is shown for a naive method that is computationally efficient. Literature in the field generally report $1\% - 5\%$ CV.

C. Line and Tree Network Performance

Figure 10 shows the sensitivity of the line and tree networks under different simulation parameters. Both networks are of length 100 nodes, the tree was generated using the method in [9].

From Figure 10(a) we see that the required sensor density decreases quite quickly vs. $P^{target}$. In an ideal line network with extremely high forecast accuracy ($\kappa = 1\%$), 1 or 2 sensors are required for extremely low missed detection errors. This extreme situation does not occur in practice, but serves as a baseline for realistic networks.

Figure 10(a) shows the sensor density vs. $\kappa$ in the line network. For all error target values except $P^{target} = 1\%$, the

D. Line and Tree Comparison

The line network is an ideal network for the method proposed in this work, since the tradeoff between $\kappa$ and $SD$ is more pronounced. This is evident in Figure 10(c) where the contours of sensor density reduce very gradually given the error target. On the other hand the tree network, for moderate forecast errors and target probabilities requires much higher measurement density.

The difficulty arising from tree graphs as opposed to line graphs is illustrated clearly in Figure 11. On average the line network requires 40% fewer sensors than the tree network. In comparison between the line and tree networks, there are two competing effects that influence the sensor density: (1) locality of flow measurements (2) hypothesis reduction in large networks.

To illustrate these factors, we examine example 3-node line and tree networks (cf. Figure 11(a)). Considering single line outages, the line network has the following possible flow measurements, $s(h_1) = 0, s(h_2) = L_1, s(h_3) = L_1 + L_2, s(h_4) = L_1 + L_2 + L_3$. Likewise the tree network has $s(h_1) = 0, s(h_2) = L_1 + L_2, s(h_3) = L_1 + L_3, s(h_4) = L_1 + L_2 + L_3$. Assuming $L_1 = 1$ and $\sigma_i^2 = 1, \forall i$, the conditional pdf of the flow measurement under each hypothesis is plotted in 11(b).
**Locality of flow measurements** Clearly, in the line network, all the hypotheses are distinguishable, whereas in the tree network $h_2$ and $h_3$ are not distinguishable. The placement algorithm will then place a sensor at $e_2$ or $e_3$ in the tree network. The main cause of this increase in placement is that flow is local (1) to the branch in the network. Therefore tree topologies will suffer from an increase in required measurements as opposed to line topologies. In the case of true feeder this can be assuaged by the fact that the mean loads on different branches can vary by very large quantities. However, if the branch in the network are sets of individual home, where we have high tree width and very similar loads, every edge will have to be monitored.

**Hypothesis reduction in large networks** When the tree is large and large number of downstream measurements have already been placed in an area network, a tree network ends up having high distinguishability. Consider a tree Figure 3 with many downstream already monitored. In this situation, many different hypothesis correspond to any number of the $2^k$ measurements which are downstream of the root. Therefore, this information is used in providing full observability even in large systems.

**E. Effect of Statistical Spread of Nominal Loads**

In the analysis in Section X-C the load distribution is fixed to the parameters determined in [13]. However to investigate the effect of the spread of the signal means in a systematic way, we repeat the method in Algorithm 3 This time however $L_{i,mc} \sim \text{U}(1, \tau)$ but with $\tau$ as a free parameter specifying the spread of the means.

However Figure 12 show that $\tau$ effects the required sensor density only in the low noise regime. Figure 12(a) and 12(b) show the effect of statistical spread of the mean for $\kappa = 1\%$. In the line network, increasing spread will increase the required sensor density. This occurs because increasing the variability of the means in the line network will increase the likelihood of overlapping likelihood functions. Notice that nominal density is 0.01 with low density, which corresponds to a single root sensor. The tree network behaves quite differently, whereby increase spread decreases the sensor density. This corresponds well with the intuition presented in Section X-D. Increasing the variation decreases the chance of fully overlapping likelihood functions.

At a moderate forecast accuracy $\kappa = 10\%$, the spread of the mean loads has much less effect. This is understandable since the spread of the likelihood function generally washes out any movement of the means. The effect is seen in both the line and tree case and is shown in Figure 12(c) and Figure 12(d).

**F. Missed Detection Error**

The optimization (5) is meant to minimize the maximum missed detection error among all possible hypothesis. This is clearly can be too conservative of a requirement. Therefore it is useful to understand the nature of the actual hypothesis missed detection values that arise from a given sensor placement.

Figure 13(a) shows the distribution of missed detection probabilities for a tree network. Setting $P_{\text{target}} = 0.2$, and $\kappa = 0.3$ we record the value of each hypothesis error. The empirical maximum error is close to the target 0.2. This makes sense because in successively solving the feasibility problem, we will expand the area network until the maximum error surpasses $P_{\text{target}}$. However, we see that in fact almost all of the missed detection probabilities are less than the target. For this example in particular 34% of the hypothesis are less than $1e^{-3}$ therefore essentially zero.

In comparing the mean and maximum errors for the range of possible values of $\kappa$ and $P_{\text{target}}$. The mean error is in on average
25% of the $P_{\text{target}}$. The maximum error in the network and $P_{\text{target}}$ very closely, therefore the optimization yields a very tight result.

\section{G. Distribution System Case Study}

We perform outage detection using a subset of the Pacific Northwest National Laboratory test feeders. The set of published feeders \cite{10} represent distribution systems at a voltage level of 4.5 KV to 35 KV. Table \ref{table:feeders} gives overview of the feeders chosen for the simulation study. The primary applications of the feeders, are heavy to light urban networks, as well as suburban and rural networks. Heavy and light industrial loads are also represented in the total set of feeders. The climate zones refer to (1) temperate (2) hot/arid (3) cold (4) hot/cold (5) hot/humid according to \cite{10}.

\begin{table}[h]
\begin{tabular}{|c|c|c|c|}
\hline
Network & Voltage & Climate Zone & Type & Size \\
\hline
R1-12.47-1 & 12.5 kV & 1 & suburban & 613 \\
R2-12.47-3 & 12.47 kV & 2 & urban & 52 \\
R5-12.47-1 & 13.8 kV & 5 & urban & 265 \\
R5-12.47-4 & 12.47 kV & 5 & commercial & 643 \\
R5-25.00-1 & 22.9 kV & 5 & suburban & 946 \\
\hline
\end{tabular}
\end{table}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure14}
\caption{Outage detection performance for selected PNNL feeders. Square marker denotes specified error target for optimization. Circular marker denotes empirical mean missed detection error for hypothesis set.}
\end{figure}

\section{XI. Conclusion}

We propose an outage detection framework combining power flow measurements on edges of the distribution system along with consumption forecasts at nodes of the network. We formulate the detection problem and provide an optimal placement for the maximum missed detection error metric. Finally, relying on feeder information from the pacific northwest national labs as well as a forecast error scaling law derived from Pacific Gas and Electric smart meter data, we demonstrate our formulation.

\section{Appendix A}

\subsection{Line and Tree Optimality}

\section*{A. Subtree Optimization}

For the optimization proposed is \cite{5} There are no guarantees of there existing any unique solution. However, we can look at a subset of this optimization of the form:

\begin{align}
\mathcal{M}(i) & = \arg \min_{\mathcal{M}} M \\
& \text{s.t. } \exists \mathcal{M}, |\mathcal{M}| \leq M \\
& P_{\text{err}}^\text{max}(\mathcal{T}_i, \mathcal{M}) \leq P_{\text{target}}
\end{align}

Where we output only the minimizing sensor placement for a subs tree $\mathcal{T}_i$. All upstream nodes are not considered. For this subproblem, we can characterize the existence of a unique optimal solution to $\mathcal{M}(i)$, albeit only for a specified set of subtrees $\mathcal{T}_i$.

\begin{proposition}
$\mathcal{M}^*$ is the unique solution of optimization \cite{5}
\iff
$P_{\text{err}}^\text{max}(\mathcal{T}_i, \mathcal{M}^*) < P_{\text{target}}$ and $\forall \mathcal{M}'$ where $|\mathcal{M}'| = |\mathcal{M}^*|$ and $\mathcal{M}' \neq \mathcal{M}^*$ we have that $P_{\text{err}}^\text{max}(\mathcal{T}_i, \mathcal{M}') > P_{\text{target}}$.
\end{proposition}

So if $\mathcal{M}^*$ is the unique minimizer, there can exist no other solution of the same size satisfying the maximum missed detection probability constraint.
B. Characterizing Placement Algorithm

Placement Algorithm method will output a solution $\mathcal{M}^g = \{m_1, \ldots, m_M\}$ in a topological order (bottom up). This set of placement can be used to construct two sets $T$ and $M$ where $T = \{t_1, \ldots, t_M\}$ is a set of tree root indices and $M = \{M_1, \ldots, M_M\}$ is a set of placements. Here $t_k = m_k$ is the $k^{th}$ sensor placed. Also, $M_k = \{m_1, \ldots, m_k\}$ the following holds $M_0^g \subseteq M_1^g \subseteq \cdots \subseteq M_M^g$. We can use $T$ and $M$ to show that Algorithm solves optimization.

Lemma 3: The sets $T$ and $M$ that are the output from Algorithm are the set of unique minimizers of optimization. Specifically $M(t_i) = M_i^g \quad \forall \ k$.

Proof: This can be shown inductively.

Base Case: Consider $t_1$ and $M_1^g$ (a single sensor at the root) which is the first constructed placement. This by construction satisfies $M(t_1) = M_1^g$ since at least one sensor must be placed this is optimal.

Inductive Hypothesis: Consider subtree $t_k$ with and placement with $M_k, M_k^g = M(t_k)$. The optimal-greedy method will move up to the root such a sensor is placed only when it can move up no further. This point is $t_{k+1}$ and the eventual placement is $M_k \cup t_{k+1}$. This is therefore optimal for $M(t_{k+1})$.

In comparison to the optimal solution to $\mathcal{M}^*$ we can then show that the solution size of the greedy method for the complete graph will be $|\mathcal{M}^*| \leq |\mathcal{M}^g(i_{	ext{root}})| \leq |\mathcal{M}^*| + 1$. To see why this is true, consider the situation where the greedy method places a sensor at the root of the tree then terminates. Then the greedy solution is the unique solution. Else it can move up to a virtual node much higher than the root and must stop there.

In the case of the line network we have the following:

Lemma 4: For a line network, $\mathcal{M}^* = |\mathcal{M}|$

Proof: Suppose Algorithm returns a solution $\mathcal{M} = \{m_1, \ldots, m_M\}$. Placement $m_k$ is arranged such that $m_1$ is the root measurement and $m_k$ is the $k^{th}$ farthest sensor from the root. Suppose some other solution $\mathcal{M}'$, the following must be true.

1) $m_1 = 1$ and $m'_1 = 1$
2) $m_k \leq m'_k \leq m_{k+1} \quad \forall k > 1$

Item (1) is true from the construction of the problem. Regarding item (2): if $m_k \geq m'_k$ the solution is no better than $\mathcal{M}$ so $m_k \leq m'_k$. Now if we have $m_k \leq m_{k+1} \leq m'_k$ this is impossible due to the increasing nature of maximum missed detection error. Since the solutions $\mathcal{M}$ and $\mathcal{M}'$ they must be of the same size.

References

[1] J.R. Aguero and A. Vargas. Inference of operative configuration of distribution networks using fuzzy logic techniques - part i: real-time model. IEEE Transactions on Power Systems, 20(3):1551–1561, 2005.
[2] R. Balakrishnan and A. Pahwa. A computer assisted intelligent storm outage evaluator for power distribution systems. IEEE Transactions on Power Delivery, 5(3):1591–1597, 1990.
[3] Christopher Bishop et al. Pattern recognition and machine learning, volume 1. Springer New York, 2006.
[4] C. Fukui and J. Kawakami. An expert system for fault section estimation using information from protective relays and circuit breakers. IEEE Transactions on Power Delivery, 1(4):83–90, 1986.
[5] M. Kezunovic. Smart fault location for smart grids. IEEE Transactions on Smart Grid, 2(1):11–22, 2011.
[6] Y. Liu and N.N. Schulz. Knowledge-based system for distribution system outage locating using comprehensive information. IEEE Transactions on Power Systems, 17(2):451–456, 2002.
[7] S.T. Mak and N. Farah. Synchronizing scada and smart meters operation for advanced smart distribution grid applications. Proc. IEEE PES Innovative Smart Grid Technologies (ISGT), pages 1–7, 2012.
[8] R.A.F. Pereira, L.G.W. da Silva, M. Kezunovic, and J.R.S. Mantovani. Improved fault location on distribution feeders based on matching during-fault voltage sags. IEEE Transactions on Power Delivery, 24(2):852–862, 2009.
[9] Alexey Rodionov and Hyunseung Choo. On generating random network structures: Trees. In Computational Science?ICCS 2003, pages 879–887. Springer, 2003.
[10] Kevin Schneider, Yousu Chen, David Chassin, Robert Pratt, David Engel, and Sandra Thompson. Modern grid initiative distribution taxonomy final report. arXiv preprint arXiv:1404.0058, 2014.
[11] Raffi Sevlian and Ram Rajagopal. Value of aggregation in smart grids. In Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on, pages 714–719. IEEE, 2013.
[12] Raffi Sevlian and Ram Rajagopal. Short term electricity load forecasting on varying levels of aggregation. arXiv preprint arXiv:1404.0058, 2014.
[13] Raffi Sevlian, Patel Siddharth, and Ram Rajagopal. Day ahead electricity load forecasting on varying levels of aggregation. Power Engineering Letters, 2014.
[14] D. Srinivasan, R.L. Cheu, Y.P. Poh, and A.K.C. Ng. Automated fault detection in power distribution networks using a hybrid fuzzy–genetic algorithm approach. Elsevier Engineering Applications of Artificial Intelligence, 13(4):407–418, 2000.
[15] Z. Sumic and R. Vidyamand. Fuzzy set theory based outage determination. Proc. of 1996 International Conference on Intelligent Systems Applications to Power Systems, pages 204–208, 1996.
[16] J. Zhu, D.L. Lubkeman, and A.A. Girgis. Automated fault location and diagnosis on electric power distribution feeders. IEEE Transactions on Power Delivery, 12(2):801–809, 1997.