GridWatch: Sensor Placement and Anomaly Detection in the Electrical Grid

Bryan Hooi[1,2]✉, Dhivya Eswaran[1], Hyun Ah Song[1], Amritanshu Pandey[3], Marko Jereminov[3], Larry Pileggi[3], and Christos Faloutsos[1]

[1] School of Computer Science, Carnegie Mellon University
[2] Department of Statistics, Carnegie Mellon University
[3] Deptartment of Electrical and Computer Engineering, Carnegie Mellon University
{bhooi,deswaran,hyunahs,christos}@cs.cmu.edu,{amritanp,mjeremin,pileggi}@andrew.cmu.edu

Abstract. Given sensor readings over time from a power grid consisting of nodes (e.g. generators) and edges (e.g. power lines), how can we most accurately detect when an electrical component has failed? More challengingly, given a limited budget of sensors to place, how can we determine where to place them to have the highest chance of detecting such a failure? Maintaining the reliability of the electrical grid is a major challenge. An important part of achieving this is to place sensors in the grid, and use them to detect anomalies, in order to quickly respond to a problem. Our contributions are: **1) Online anomaly detection:** we propose a novel, online anomaly detection algorithm that outperforms existing approaches. **2) Sensor placement:** we construct an optimization objective for sensor placement, with the goal of maximizing the probability of detecting an anomaly. We show that this objective has the property of submodularity, which we exploit in our sensor placement algorithm. **3) Effectiveness:** Our sensor placement algorithm is provably near-optimal, and both our algorithms outperform existing approaches in accuracy by 59% or more (F-measure) in experiments. **4) Scalability:** our algorithms scale **linearly**, and our detection algorithm is **online**, requiring bounded space and constant time per update.

1 Introduction

Improving the efficiency and security of power delivery is a critically important goal, in the face of disturbances arising from severe weather, human error, equipment failure, or even intentional intrusion. Estimates [5] suggest that reducing outages in the U.S. grid could save $49 billion per year, reduce emissions by 12 to 18%, while improving efficiency could save an additional $20.4 billion per year. A key part of achieving this goal is to use sensor monitoring data to quickly identify when parts of the grid fail, so as to quickly respond to the problem.

A major challenge is scalability - power systems data can be both high-volume and received in real time, since the data comes from sensors which are continuously monitoring the grid. This motivates us to develop fast methods

that work in this online (or streaming) setting. When each new data point is received, the algorithm should update itself efficiently.

Hence, our goal is an online anomaly detection algorithm:

Informal Problem 1 (Online Anomaly Detection)**.**
- ***Given:*** *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a subset $\mathcal{S}$ of nodes which contain sensors. For each sensor, we have a continuous stream of values of real and imaginary voltage $V(t)$ and current $I(t)$ measured by these sensors.*
- ***Find:*** *At each time t, compute an anomalousness score $A(t)$, indicating our confidence level that an anomaly occurred (i.e. a transmission line failed).*

For cost reasons, it is generally infeasible to place sensors at every node. Hence, an important follow-up question is where to place sensors so as to maximize the probability of detecting an anomaly.

Informal Problem 2 (Sensor Placement)**.**
- ***Given:*** *A budget k of the number of sensors we can afford, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a simulator that allows us to simulate sensor readings at each node.*
- ***Find:*** *A set of nodes $\mathcal{S} \subseteq \mathcal{V}$, which are the locations we should place our sensors, such that $|\mathcal{S}| = k$.*

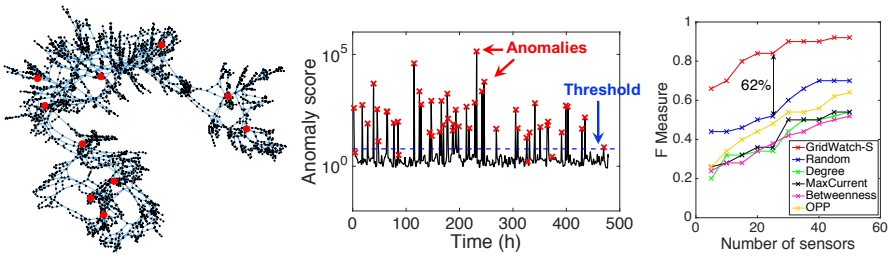
In contrast to most approaches, our anomaly detection algorithm, GridWatch-D, uses a domain-dependent approach which exploits the fact that electrical sensors consist of a voltage reading at a node as well as the current along each adjacent edge. This allows us to detect anomalies more accurately, even when using an online approach. Next, we propose GridWatch-S, a sensor placement algorithm. The main idea is to define an objective which estimates our probability of successfully detecting an anomaly, then show that this objective has the submodularity property, allowing us to optimize it with approximation guarantees using an efficient greedy algorithm.

Figure 1a shows the sensors selected by GridWatch-S: red circles indicate positions chosen. Figure 1b shows the anomaly scores (black line) output by GridWatch-D, which accurately match the ground truth. Figure 1c shows that GridWatch-S outperforms baselines on the case2869 data.

Our contributions are as follows:

1. **Online anomaly detection:** we propose a novel, online anomaly detection algorithm, GridWatch-D, that outperforms existing approaches.
2. **Sensor placement:** we construct an optimization objective for sensor placement, with the goal of maximizing the probability of detecting an anomaly. We show that this objective has the property of 'submodularity,' which we exploit to propose our sensor placement algorithm.
3. **Effectiveness:** Our sensor placement algorithm, GridWatch-S, is provably near-optimal. In addition, both our algorithms outperform existing approaches in accuracy by 59% or more (F-measure) in experiments.
4. **Scalability:** Our algorithms scale linearly, and GridWatch-D is online, requiring bounded space and constant time per update.

Reproducibility: our code and data are publicly available at `github.com/bhooi/gridwatch`.



(a) **Sensor selection** (b) **Anomaly scores** (c) **Accuracy**

Fig. 1: (a) GridWatch-S provably selects near-optimal sensor locations. Red circles indicate positions chosen for sensors, in the case2869 graph. (b) GridWatch-D computes anomaly scores (black line) on case2869. Red crosses indicate ground truth - notice 100% true alarms (all black spikes above blue line are true alarms) and only 4 false dismissals (red crosses below blue line). (c) F-measure of GridWatch-S compared to baselines on case2869.

2 Background and Related Work

Time Series Anomaly Detection. Numerous algorithms exist for anomaly detection in univariate time series [17]. For multivariate time series, LOF [8] uses a local density approach. Isolation Forests [20] partition the data using a set of trees for anomaly detection. Other approaches use neural networks [34], distance-based [28], and exemplars [15]. However, none of these consider sensor selection.

Anomaly Detection in Temporal Graphs. [4] finds anomalous changes in graphs using an egonet (i.e. neighborhood) based approach, while [10,22] uses community-based approaches. [3] finds change points in dynamic graphs, while other partition-based [2] and sketch-based [29] exist for anomaly detection. However, these methods require fully observed edge weights (i.e. all sensors present), and do not consider sensor selection.

Power Grid Monitoring. A number of works consider the Optimal PMU Placement (OPP) problem [9], of optimally placing sensors in power grids, typically to make as many nodes as possible fully observable, or minimizing mean-squared error. Greedy [19], convex relaxation [16], integer program [12], simulated annealing [7] have been proposed. However, these do not perform anomaly detection. [27,36,21] consider OPP in the presence of branch outages, but not anomalies in general, and due to their use of integer programming, only use small graphs of size at most 60.

Epidemic and Outbreak Detection. [18] proposed CELF, for outbreak detection in networks, such as water distribution networks and blog data, also using a submodular objective function. Their setting is a series of cascades spreading over the graph, while our input data is time-series data from sensors at various edges of the graph. For epidemics, [25,11] consider targeted immunization, such as identifying high-degree [25] or well-connected [11] nodes. We show experimentally that our sensor selection algorithm outperforms both approaches.

Table 1: Comparison of related approaches: only GridWatch satisfies all the listed properties.

	Temporal [17,8], etc.	**Graph-based** [4,22,30]	**OPP** [9,19,16], etc.	**Immunize** [25,11]	**GridWatch**
Anomaly Detection	✓	✓			✔
Online Algorithm	✓				✔
Using Graph Data		✓	✓	✓	✔
Sensor Selection			✓	✓	✔
Approx Guarantee					✔

Table 1 summarizes related work. GridWatch differs from existing methods in that it performs anomaly detection using an online algorithm, and it selects sensor locations with a provable approximation guarantee.

2.1 Background: Submodular Functions

A function f defined on subsets of $\mathcal{V}$ is submodular if whenever $\mathcal{T} \subseteq S$ and $i \notin S$:

$$f(\mathcal{S} \cup \{i\}) - f(\mathcal{S}) \leq f(\mathcal{T} \cup \{i\}) - f(\mathcal{T}) \tag{1}$$

Intuitively, this can be interpreted as *diminishing returns*: the left side is the gain in f from adding i to $\mathcal{S}$, and the right side is the gain from adding i to $\mathcal{T}$. Since $\mathcal{T} \subseteq S$, this says that as $\mathcal{T}$ 'grows' to $\mathcal{S}$, the gains from adding i can only diminish.

[23] showed that nondecreasing submodular functions can be optimized by a greedy algorithm with a constant-factor approximation guarantee of $(1 - 1/e)$. These were extended by [32] to the non-constant sensor cost setting.

3 GridWatch-D Anomaly Detection Algorithm

Preliminaries Table 2 shows the symbols used in this paper.

In this section, we are given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a fixed set of sensors $\mathcal{S} \subseteq \mathcal{V}$. Each sensor consists of a central node i on which voltage $V_i(t) \in \mathbb{C}$

Table 2: Symbols and definitions

Symbol	Interpretation
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Input graph
$\mathcal{S}$	Subset of nodes to place sensors on
n	Number of nodes
s	Number of scenarios
$\mathcal{N}_i$	Set of edges adjacent to node i
$V_i(t)$	Voltage at node i at time t
$I_e(t)$	Current at edge e at time t
$S_{ie}(t)$	Power w.r.t. node i and edge e at time t
$\Delta S_{ie}(t)$	Power change: $\Delta S_{ie}(t) = S_{ie}(t) - S_{ie}(t-1)$
$X_i(t)$	Sensor vector for scenario i at time t
c	Anomalousness threshold parameter
$\tilde{\mu}_i(t)$	Median of sensor i at time t
$\tilde{\sigma}_i(t)$	Inter-quartile range of sensor i at time t
$a_i(t)$	Sensor-level anomalousness for sensor i at time t
$A(t)$	Total anomalousness at time t

is measured, at each time t. Note that complex voltages and currents are used to take phase into account, following standard practice in circuit analysis (this paper will not presume familiarity with this). Additionally, for sensor i, letting $\mathcal{N}_i$ be the set of edges adjacent to i, we are given the current $I_e \in \mathbb{C}$ along each edge $e \in \mathcal{N}_i$.

For sensor i and edge $e \in \mathcal{N}_i$, define the power w.r.t. i along edge e as $S_{ie}(t) = V_i(t) \cdot I_e(t)^*$, where * is the complex conjugate. We find that using power (rather than current) provides better anomaly detection in practice. However, when considering the edges around a single sensor i, variations in current result in similar variations in power, so they perform the same role.

3.1 Types of Anomalies

Our goal is to detect single edge deletions, i.e. a transmission line failure. Single edge deletions affect the voltage and current in the graph in a complex, nonlinear way, and can manifest themselves in multiple ways. Consider the illustrative power grid shown by the graphs in Figure 2. The power grid consists of a single generator, a single load, and power lines of uniform resistance. When the edge marked in the black cross fails, current is diverted from some edges to others, causing some edges to have increased current flow (blue edges), and thus increased power, and others to have decreased current flow (red edges). Current flows are computed using a standard power grid simulator, Matpower [37].

In the leftmost plot, the edge deletion diverts a large amount of current into a single edge, resulting in a highly anomalous value (+0.4) along a single edge. To detect single-edge anomalies, we consider the largest absolute change in power in the edges adjacent to this sensor. Formally, letting $\Delta S_{ie}(t) = S_{ie}(t) - S_{ie}(t-1)$,

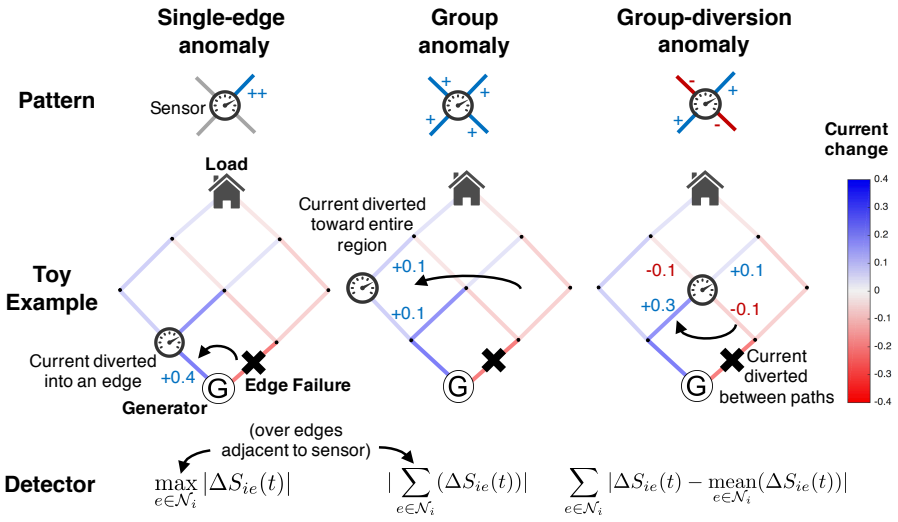


Fig. 2: **Domain-aware model for anomalies:** edge failures form 3 patterns. Edge color indicates change in current due to the edge failure: blue is an increase; red is a decrease. *Left:* diversion into a single edge. *Center:* diversion from the right to the left side of the graph, forming a group anomaly. *Right:* diversion between paths, forming a group diversion anomaly.

Definition 1 (Single-Edge Detector). *The detector at sensor i is:*

$$x_{SE,i}(t) = \max_{e\in\mathcal{N}_i} |\Delta S_{ie}(t)| \tag{2}$$

In the middle plot, the edge deletion cuts off a large amount of current that would have gone from the generator toward the right side of the graph, diverting it into the left side of the graph. This results in some nodes in the left region with all their neighboring edges having positive changes (blue), such as the leftmost node. Individually, these changes may be too small to appear anomalous, but in aggregate, they provide stronger evidence of an anomaly. Hence, the group anomaly detector computes the sum of power changes around sensor i, then takes the absolute value:

Definition 2 (Group Anomaly Detector). *The detector at sensor i is:*

$$x_{GA,i}(t) = |\sum_{e\in\mathcal{N}_i} (\Delta S_{ie}(t))| \tag{3}$$

In the right plot, the edge deletion diverts current between nearby edges. In particular, current diversions around the central node cause it to have neighbors which greatly differ from each other: 2 positive edges and 2 negative edges. If this diversion is large enough, this provides stronger evidence of an anomaly than simply looking at each edge individually. Hence, the group diversion detector

measures the 'spread' around sensor i by looking at the total absolute deviation of power changes about sensor i:

Definition 3 (Group Diversion Detector). *The detector at sensor i is:*

$$x_{GD,i}(t) = \sum_{e \in \mathcal{N}_i} |\Delta S_{ie}(t) - \underset{e \in \mathcal{N}_i}{mean}(\Delta S_{ie}(t))| \tag{4}$$

3.2 Proposed Anomaly Score

Having computed our detectors, we now define our anomaly score. For each sensor i, concatenate its detectors into a vector:

$$X_i(t) = [x_{SE,i}(t) \;\; x_{GA,i}(t) \;\; x_{GD,i}(t)] \tag{5}$$

Sensor i should label time t as an anomaly if any of the detectors greatly deviate from their historical values. Hence, let $\tilde{\mu}_i(t)$ and $\tilde{\sigma}_i(t)$ be the historical median and inter-quartile range (IQR)[4] [35] of $X_i(t)$ respectively: i.e. the median and IQR of $X_i(1), \cdots, X_i(t-1)$. We use median and IQR generally instead of mean and standard deviation as they are robust against anomalies, since our goal is to detect anomalies.

Thus, define the sensor-level anomalousness as the maximum number of IQRs that any detector is away from its historical median. The infinity-norm $\|\cdot\|_\infty$ denotes the maximum absolute value of a vector.

Definition 4 (Sensor-level anomalousness). *Sensor-level anomalousness is:*

$$a_i(t) = \left\| \frac{X_i(t) - \tilde{\mu}_i(t)}{\tilde{\sigma}_i(t)} \right\|_\infty \tag{6}$$

Finally, the **overall anomalousness** at time t is the maximum of $a_i(t)$ over all sensors. Taking maximums allows us to determine the *location* (not just time) of an anomaly, by looking at which sensor contributed toward the maximum.

Definition 5 (Overall anomalousness). *Overall anomalousness at time t is:*

$$A(t) = \max_{i \in \mathcal{S}} a_i(t) \tag{7}$$

Algorithm 1 summarizes our GridWatch-D anomaly detection algorithm. Note that we can maintain the median and IQR of a set of numbers in a streaming manner using reservoir sampling [33]. Hence, the Normalize operation in Line 5 takes a value of $\Delta S_{ie}(t)$, subtracts its historical median and divides by the historical IQR for that sensor. This ensures that sensors with large averages or spread do not dominate.

Lemma 1. GridWatch-D *is online, and requires bounded memory and time.*

Proof. We verify from Algorithm 1 that GridWatch-D's memory consumption is $O(|\mathcal{S}|)$, and updates in $O(|\mathcal{S}|)$ time per iteration, which are bounded (regardless of the length of the stream). ■

[4] IQR is a robust measure of spread, equal to the difference between the 75% and 25% quantiles.

Algorithm 1: GridWatch-D online anomaly detection algorithm

Input : Graph $\mathcal{G}$, voltage $V_i(t)$, current $I_i(t)$
Output: Anomalousness score $A(t)$ for each t, where higher $A(t)$ indicates greater certainty of an anomaly

for *t received as a stream:* **do**
  for $i \in \mathcal{S}$ **do**
    $S_{ie}(t) \leftarrow V_i(t) \cdot I_e^*(t) \; \forall \; e \in \mathcal{N}_i$ ▷Power
    $\Delta S_{ie}(t) \leftarrow S_{ie}(t) - S_{ie}(t-1)$ ▷Power differences
    $\Delta S_{i\cdot}(t) \leftarrow \text{Normalize}(\Delta S_{i\cdot})$
    Compute detectors $x_{SE,i}(t)$, $x_{GA,i}(t)$ and $x_{GD,i}(t)$ using Eq. (2) to (4)
    Concatenate detectors: $X_i(t) = [x_{SE,i}(t) \;\; x_{GA,i}(t) \;\; x_{GD,i}(t)]$
    $\tilde{\mu}_i(t) \leftarrow \text{UpdateMedian}(\tilde{\mu}_i(t-1), X_i(t))$ ▷Historical median
    $\tilde{\sigma}_i(t) \leftarrow \text{UpdateIQR}(\tilde{\sigma}_i(t-1), X_i(t))$ ▷Historical IQR
    $a_i(t) \leftarrow \| \frac{X_i(t) - \tilde{\mu}_i(t)}{\tilde{\sigma}_i(t)} \|_\infty$ ▷Sensor-level anomalousness
  $A(t) = \max_{i \in \mathcal{S}} a_i(t)$ ▷Overall anomalousness

4 Sensor Placement: GridWatch-S

So far, we have detected anomalies using a fixed set of sensors. We now consider how to select locations for sensors to place given a fixed budget of k sensors to place. Our main idea will be to construct an optimization objective for the anomaly detection performance of a subset $\mathcal{S}$ of sensor locations, and show that this objective has the 'submodularity' property, showing that a greedy approach gives approximation guarantees.

Note the change in problem setting: we are no longer monitoring for anomalies online in time series data, since we are now assuming that the sensors have not even been installed yet. Instead, we are an offline planner deciding where to place the sensors. To do this, we use a model of the system in the form of its graph $\mathcal{G}$, plugging it into a simulator such as Matpower [37] to generate a dataset of ground truth anomalies and normal scenarios, where the former contain a randomly chosen edge deletion, and the latter do not.

4.1 Proposed Optimization Objective

Intuitively, we should select sensors $\mathcal{S}$ to maximize the **probability of detecting an anomaly**. This probability can be estimated as the fraction of ground truth anomalies that we successfully detect. Hence, our optimization objective, $f(\mathcal{S})$, will be the fraction of anomalies that we successfully detect when using GridWatch-D, with sensor set $\mathcal{S}$. We will now formalize this and show that it is submodular.

Specifically, define $X_i(r)$ as the value of sensor i on the rth anomaly, analogous to (5). Also define $\tilde{\mu}_i$ and $\tilde{\sigma}_i$ as the median and IQR of sensor i on the full set of normal scenarios. Also let $a_i(r)$ be the sensor-level anomalousness of the

rth anomaly, which can be computed as in Definition 4 plugging in $\tilde{\mu}_i$ and $\tilde{\sigma}_i$:

$$a_i(r) = \left\| \frac{X_i(r) - \tilde{\mu}_i}{\tilde{\sigma}_i} \right\|_\infty \tag{8}$$

Define overall anomalousness w.r.t. $\mathcal{S}$, $A(r, \mathcal{S})$, analogously to Definition 5:

$$A(r, \mathcal{S}) = \max_{i \in \mathcal{S}} a_i(r) \tag{9}$$

Given threshold c, anomaly r will be detected by sensor set $\mathcal{S}$ if and only if $A(r, \mathcal{S}) > c$. Hence, our optimization objective is to maximize the fraction of detected anomalies:

$$\underset{\mathcal{S} \subseteq \mathcal{V}, |S| = k}{\text{maximize}} \;\; f(\mathcal{S}), \text{ where } f(\mathcal{S}) = \frac{1}{s} \sum_{r=1}^{s} \mathbf{1}\{A(r, \mathcal{S}) > c\} \tag{10}$$

4.2 Properties of Objective

Our optimization objective $f(\mathcal{S})$ is submodular: informally, it exhibits diminishing returns. The more sensors we add, the smaller the marginal gain in detection probability.

Theorem 1. *Detection probability $f(\mathcal{S})$ is submodular, i.e. for all subsets $\mathcal{T} \subseteq \mathcal{S}$ and nodes $i \in \mathcal{V} \setminus \mathcal{S}$:*

$$f(\mathcal{S} \cup \{i\}) - f(\mathcal{S}) \leq f(\mathcal{T} \cup \{i\}) - f(\mathcal{T}) \tag{11}$$

Proof.

$$\begin{aligned}
f(\mathcal{S} \cup \{i\}) - f(\mathcal{S}) &= \frac{1}{s} \sum_{r=1}^{s} \left(\mathbf{1}\{A(r, \mathcal{S} \cup \{i\}) > c\} - \mathbf{1}\{A(r, \mathcal{S}) > c\} \right) \\
&= \frac{1}{s} \sum_{r=1}^{s} \left(\mathbf{1}\{ \max_{j \in \mathcal{S} \cup \{i\}} a_j(r) > c\} - \mathbf{1}\{\max_{j \in \mathcal{S}} a_j(r) > c\} \right) \\
&= \frac{1}{s} \sum_{r=1}^{s} \left(\mathbf{1}\{a_i(r) > c \wedge \max_{j \in \mathcal{S}} a_j(r) \leq c\} \right) \\
&\leq \frac{1}{s} \sum_{r=1}^{s} \left(\mathbf{1}\{a_i(r) > c \wedge \max_{j \in \mathcal{T}} a_j(r) \leq c\} \right) \\
&= f(\mathcal{T} \cup \{i\}) - f(\mathcal{T})
\end{aligned}$$

■

Theorem 2. *$f(\mathcal{S})$ is nondecreasing, i.e. $f(\mathcal{T}) \leq f(\mathcal{S})$ for all subsets $\mathcal{T} \subseteq \mathcal{S}$.*

Proof.

$$f(\mathcal{S}) = \frac{1}{s} \sum_{r=1}^{s} A(r, \mathcal{S}) = \frac{1}{s} \sum_{r=1}^{s} \max_{j \in \mathcal{S}} a_j(r) \geq \frac{1}{s} \sum_{r=1}^{s} \max_{j \in \mathcal{T}} a_j(r) = f(\mathcal{T})$$

■

4.3 Proposed GridWatch-S Algorithm

We exploit this submodularity using an efficient greedy algorithm that starts from $\mathcal{S}$ as the empty set, and iteratively adds the best sensor to maximize $f(\mathcal{S})$, until the budget constraint $|S| = k$ is reached. Algorithm 2 describes our GridWatch-S algorithm.

Algorithm 2: GridWatch-S sensor selection algorithm

Input : Graph $\mathcal{G}$, voltage $V_i(t)$, current $I_i(t)$, budget k, sensor scores $a_i(r)$ from (8)
Output: Chosen sensor set $\mathcal{S}$

$\mathcal{S} = \{\}$
Initialize $A(r) = 0 \; \forall \; r \in \mathcal{S}$ ▷Overall anomalousness is all zero since $\mathcal{S} = \{\}$
while $|\mathcal{S}| < k$ **do**
  for $i \notin \mathcal{S}$ **do**
    $\delta_i \leftarrow \frac{1}{s} \sum_{r=1}^{s} \mathbf{1}\{\max(A(r), a_i(r)) > c\}$ ▷Objective value if we added i to $\mathcal{S}$
  $i^* \leftarrow \arg\max_{i \notin \mathcal{S}} \delta_i$ ▷Greedily add the sensor that maximizes objective
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$
  $A(r) = \max(A(r), a_{i^*}(r)) \; \forall \; r \in \mathcal{S}$

4.4 Approximation Bound

The nondecreasing and submodularity properties of f imply that Algorithm 2 achieves at least $1 - 1/e$ ($\approx 63\%$) of the value of the optimal sensor placement. Letting $\hat{\mathcal{S}}$ be the set returned by Algorithm 2, and $\mathcal{S}^*$ be the optimal set:

Theorem 3.

$$f(\hat{\mathcal{S}}) \geq (1 - 1/e) f(S^*) \tag{12}$$

Proof. This follows from [23] since f is nondecreasing and submodular. ■

5 Experiments

We design experiments to answer the following questions:

- **Q1. Anomaly Detection Accuracy:** on a fixed set of sensors, how accurate are the anomalies detected by GridWatch-S compared to baselines?
- **Q2. Sensor Selection:** how much does sensor selection using GridWatch-S improve the anomaly detection performance compared to baselines?
- **Q3. Scalability:** how do our algorithms scale with the graph size?

Our code and data are publicly available at `github.com/bhooi/gridwatch`. Experiments were done on a 2.4 GHz Intel Core i5 Macbook Pro, 16 GB RAM running OS X 10.11.2.

Data: We use 2 graphs, case2869 and case9241, which accurately represent different parts of the European high voltage network [37]. case2869 contains 2869 nodes (generators or buses) and 2896 edges (power lines or transformers). case9241 contains 9241 nodes and 16049 edges.

5.1 Q1. Anomaly Detection Accuracy

In this section, we compare GridWatch-D against baseline anomaly detection approaches, given a fixed set of sensors.

Experimental Settings: For each graph, the sensor set for all algorithms is chosen as a uniformly random set of nodes of various sizes (the sizes are plotted in the x-axis of Figure 3). Then, out of 480 time ticks, we first sample 50 random time ticks as the times when anomalies occur. In each such time tick, we deactivate a randomly chosen edge (i.e. no current can flow over that edge).

Using MatPower [37], we then generate voltage and current readings at each sensor. This requires an input time series of loads (i.e. real and reactive power at each node): we use load patterns estimated from real data [31] recorded from the Carnegie Mellon University (CMU) campus for 20 days from July 29 to August 17, 2016, scaled to a standard deviation of $0.3 \cdot \sigma$, with added Gaussian noise of $0.2 \cdot \sigma$, where σ is the standard deviation of the original time series [31].

This results in a time series of 480 time ticks (hourly data from 20 days), at each time recording the voltage at each sensor and the current at each edge adjacent to one of the sensors. Given this input, each algorithm then returns a ranking of the anomalies. We evaluate this using standard metrics, AUC (area under the ROC curve) and F-measure ($\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$), the latter computed on the top 50 anomalies output by each algorithm.

Baselines: Dynamic graph anomaly detection approaches [4,10,22,6,30] cannot be used as they require graphs with fully observed edge weights. Moreover, detecting failed power lines with all sensors present can be done by simply checking if any edge has current equal to 0, which is trivial. Hence, instead, we compare GridWatch-D to the following multidimensional anomaly detection methods: Isolation Forests [20], Vector Autoregression (VAR) [14], Local Outlier Factor (LOF) [8], and Parzen Window [24]. Each uses the currents and voltages at the given sensors as features. For VAR the norms of the residuals are used as anomaly scores; the remaining methods return anomaly scores directly. For Isolation Forests, we use 100 trees (following the scikit-learn defaults [26]). For VAR we select the order by maximizing AIC, following standard practice. For LOF we use 20 neighbors, and 20 neighbors for Parzen Window.

Figure 3 shows that GridWatch-D outperforms the baselines, by 31% to 42% Area under the Curve (AUC) and 133% to 383% F-Measure. The gains in performance likely come from the use of the 3 domain-knowledge based detectors, which combine information from the currents surrounding each sensor in a way that makes it clearer when an anomaly occurs.

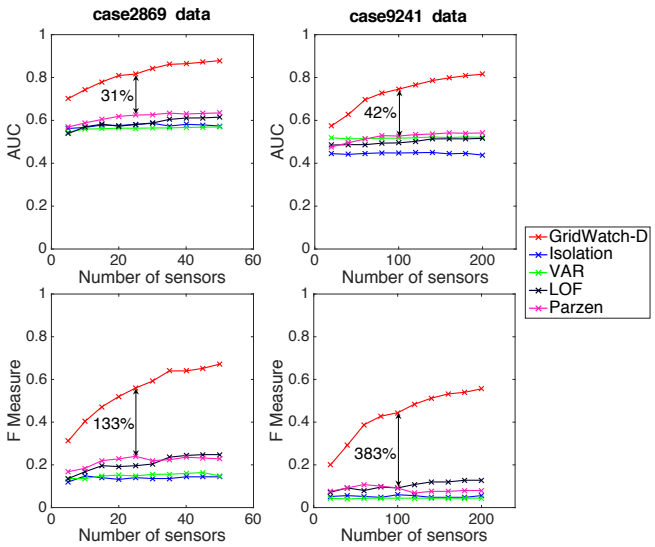


Fig. 3: **Accurate anomaly detection:** GridWatch-D outperforms the baselines. Left plots are for case2869; right plots are for case9241.

Further testing shows that GridWatch-D's 3 detectors all play a role: e.g. on case2869, for 50 sensors, GridWatch-D has F-measure 0.67, but only using single detectors 1, 2 or 3 (where detector 1 refers to the detector in Definition 1, and so on) gives F-measures of 0.51, 0.6 or 0.56 respectively.

5.2 Q2. Sensor Selection Quality

We now evaluate GridWatch-S. We use the same settings as in the previous sub-section, except that the sensors are now chosen using either GridWatch-S, or one of the following baselines. We then compute the anomaly detection performance of GridWatch-D as before on each choice of sensors. For GridWatch-S we use $c = 15$. For our simulated data sizes, we assume 2000 anomalies and 480 normal scenarios.

Baselines: randomly selected nodes (*Random*); highest degree nodes (*Degree*); nodes with highest total current in their adjacent edges (*MaxCurrent*); highest betweenness centrality [13] nodes, i.e. nodes with the most shortest paths passing through them, thus being the most 'central' (*Betweenness*); a power-grid based Optimal PMU Placement algorithm using depth-first search (*OPP* [7]).

Figure 4 shows that GridWatch-S outperforms the baselines, by 18 to 19% Area under the Curve (AUC) and 59 to 62% F-Measure.

Figure 1b shows the GridWatch-S scores on case2869 over time, when using the maximum 200 sensors, with red crosses where true anomalies exist. Spikes in anomaly score match very closely with the true anomalies.

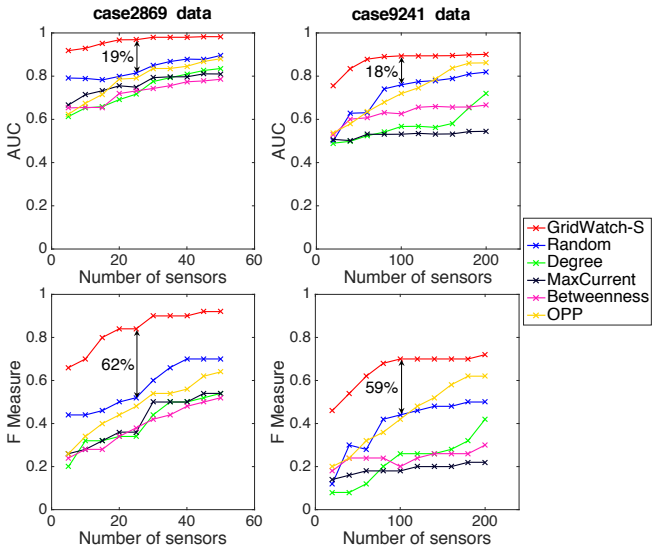
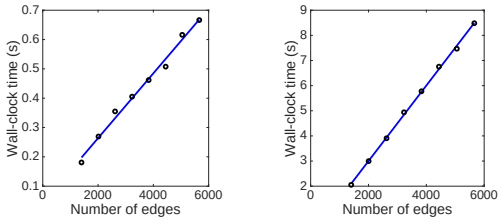


Fig. 4: **GridWatch-S provides effective sensor selection:** sensor selection using GridWatch-S results in higher detection accuracy than baselines.

5.3 Q3. Scalability

Finally, we evaluate the scalability of GridWatch-D and GridWatch-S. To generate graphs of different sizes, we start with the IEEE 118-bus network [1], which represents a portion of the US power grid in 1962, and duplicate it $2, 4, \cdots, 20$ times. To keep our power grid connected, after each duplication, we add edges from each node to its counterpart in the last duplication; the parameters of each such edge are randomly sampled from those of the actual edges. We then run GridWatch-D and GridWatch-S using the same settings as the previous sub-section. Figure 5b shows that GridWatch-D and GridWatch-S scale linearly. The blue line is the best-fit regression line.



(a) GridWatch-D (b) GridWatch-S

Fig. 5: Our algorithms scale linearly: wall-clock time of (a) GridWatch-D and (b) GridWatch-S against number of edges in $\mathcal{G}$.

6 Conclusion

In this paper, we proposed GridWatch-D, an online algorithm that accurately detects anomalies in power grid data. The main idea of GridWatch-D is to design domain-aware detectors that combine information at each sensor appropriately. We then proposed GridWatch-S, a sensor placement algorithm, which uses a submodular optimization objective. While our method could be technically applied to any type of graph-based sensor data (not just power grids), the choice of our detectors is motivated by our power grid setting. Hence, future work could study how sensitive various detectors are for detecting anomalies in graph-based sensor data from different domains.

Our contributions are as follows:

1. **Online anomaly detection:** we propose a novel, online anomaly detection algorithm, GridWatch-D that outperforms existing approaches.
2. **Sensor placement:** we construct an optimization objective for sensor placement, with the goal of maximizing the probability of detecting an anomaly. We show that this objective is submodular, which we exploit in our sensor placement algorithm.
3. **Effectiveness:** Due to submodularity, GridWatch-S, our sensor placement algorithm is provably near-optimal. In addition, both our algorithms outperform existing approaches in accuracy by 59% or more (F-measure) in experiments.
4. **Scalability:** Our algorithms scale linearly, and GridWatch-D is online, requiring bounded space and constant time per update.

Reproducibility: our code and data are publicly available at `github.com/bhooi/gridwatch`.

7 Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1314632, IIS-1408924, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, and in part by the Defense Advanced Research Projects Agency (DARPA) under award no. FA8750-17-1-0059 for the RADICS program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Ieee power systems test case archive. `http://www2.ee.washington.edu/research/pstca/`, accessed: 2017-03-15

2. Aggarwal, C.C., Zhao, Y., Philip, S.Y.: Outlier detection in graph streams. In: Data Engineering (ICDE), 2011 IEEE 27th International Conference on. pp. 399–409. IEEE (2011)
3. Akoglu, L., Faloutsos, C.: Event detection in time series of mobile communication graphs. In: Army science conference. pp. 77–79 (2010)
4. Akoglu, L., McGlohon, M., Faloutsos, C.: Oddball: Spotting anomalies in weighted graphs. In: PAKDD. pp. 410–421. Springer (2010)
5. Amin, S.M.: Us grid gets less reliable [the data]. IEEE Spectrum **48**(1), 80–80 (2011)
6. Araujo, M., Papadimitriou, S., Günnemann, S., Faloutsos, C., Basu, P., Swami, A., Papalexakis, E.E., Koutra, D.: Com2: fast automatic discovery of temporal (comet) communities. In: PAKDD. pp. 271–283. Springer (2014)
7. Baldwin, T., Mili, L., Boisen, M., Adapa, R.: Power system observability with minimal phasor measurement placement. IEEE Transactions on Power Systems **8**(2), 707–715 (1993)
8. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: ACM sigmod record. vol. 29, pp. 93–104. ACM (2000)
9. Brueni, D.J., Heath, L.S.: The pmu placement problem. SIAM Journal on Discrete Mathematics **19**(3), 744–761 (2005)
10. Chen, Z., Hendrix, W., Samatova, N.F.: Community-based anomaly detection in evolutionary networks. Journal of Intelligent Information Systems **39**(1), 59–85 (2012)
11. Cohen, R., Havlin, S., Ben-Avraham, D.: Efficient immunization strategies for computer networks and populations. Physical review letters **91**(24), 247901 (2003)
12. Dua, D., Dambhare, S., Gajbhiye, R.K., Soman, S.: Optimal multistage scheduling of pmu placement: An ilp approach. IEEE Transactions on Power delivery **23**(4), 1812–1820 (2008)
13. Freeman, L.C.: Centrality in social networks conceptual clarification. Social networks **1**(3), 215–239 (1978)
14. Hamilton, J.D.: Time series analysis, vol. 2. Princeton university press Princeton (1994)
15. Jones, M., Nikovski, D., Imamura, M., Hirata, T.: Anomaly detection in real-valued multidimensional time series. In: International Conference on Bigdata/Socialcom/Cybersecurity. Stanford University, ASE. Citeseer (2014)
16. Kekatos, V., Giannakis, G.B., Wollenberg, B.: Optimal placement of phasor measurement units via convex relaxation. IEEE Transactions on power systems **27**(3), 1521–1530 (2012)
17. Keogh, E., Lin, J., Lee, S.H., Van Herle, H.: Finding the most unusual time series subsequence: algorithms and applications. Knowledge and Information Systems **11**(1), 1–27 (2007)
18. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: KDD. pp. 420–429. ACM (2007)
19. Li, Q., Negi, R., Ilić, M.D.: Phasor measurement units placement for power system state estimation: A greedy approach. In: Power and Energy Society General Meeting, 2011 IEEE. pp. 1–8. IEEE (2011)
20. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: ICDM. pp. 413–422. IEEE (2008)
21. Magnago, F.H., Abur, A.: A unified approach to robust meter placement against loss of measurements and branch outages. In: Power Industry Computer Applications, 1999. PICA'99. Proceedings of the 21st 1999 IEEE International Conference. pp. 3–8. IEEE (1999)

22. Mongiovi, M., Bogdanov, P., Ranca, R., Papalexakis, E.E., Faloutsos, C., Singh, A.K.: Netspot: Spotting significant anomalous regions on dynamic networks. In: SDM. pp. 28–36. SIAM (2013)
23. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functionsi. Mathematical Programming **14**(1), 265–294 (1978)
24. Parzen, E.: On estimation of a probability density function and mode. The annals of mathematical statistics **33**(3), 1065–1076 (1962)
25. Pastor-Satorras, R., Vespignani, A.: Immunization of complex networks. Physical Review E **65**(3), 036104 (2002)
26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
27. Rakpenthai, C., Premrudeepreechacharn, S., Uatrongjit, S., Watson, N.R.: An optimal pmu placement method against measurement loss and branch outage. IEEE transactions on power delivery **22**(1), 101–107 (2007)
28. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: ACM Sigmod Record. vol. 29, pp. 427–438. ACM (2000)
29. Ranshous, S., Harenberg, S., Sharma, K., Samatova, N.F.: A scalable approach for outlier detection in edge streams using sketch-based approximations. In: SDM. pp. 189–197. SIAM (2016)
30. Shah, N., Koutra, D., Zou, T., Gallagher, B., Faloutsos, C.: Timecrunch: Interpretable dynamic graph summarization. In: KDD. pp. 1055–1064. ACM (2015)
31. Song, H.A., Hooi, B., Jereminov, M., Pandey, A., Pileggi, L., Faloutsos, C.: Powercast: Mining and forecasting power grid sequences. In: ECML-PKDD. pp. 606–621. Springer (2017)
32. Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. Operations Research Letters **32**(1), 41–43 (2004)
33. Vitter, J.S.: Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS) **11**(1), 37–57 (1985)
34. Yi, S., Ju, J., Yoon, M.K., Choi, J.: Grouped convolutional neural networks for multivariate time series. arXiv preprint arXiv:1703.09938 (2017)
35. Yule, G.U.: An introduction to the theory of statistics. C. Griffin, limited (1919)
36. Zhao, Y., Goldsmith, A., Poor, H.V.: On pmu location selection for line outage detection in wide-area transmission networks. In: Power and Energy Society General Meeting, 2012 IEEE. pp. 1–8. IEEE (2012)
37. Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J.: Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Transactions on power systems **26**(1), 12–19 (2011)