ABSTRACT
Motivated by the recent cyber attack on the Ukrainian power grid, we study cyber attacks on power grids that affect both the physical infrastructure and the data at the control center. In particular, we assume that an adversary attacks an area by: (i) remotely disconnecting some lines within the attacked area, and (ii) modifying the information received from the attacked area to mask the line failures and hide the attacked area from the control center. For the latter, we consider two types of attacks: (i) data distortion: which distorts the data by adding powerful noise to the actual data, and (ii) data replay: which replays a locally consistent old data instead of the actual data. We use the DC power flow model and prove that the problem of finding the set of line failures given the phase angles of the nodes outside of the attacked area is strongly NP-hard, even when the attacked area is known. However, we introduce the polynomial time REcurrent Attack Containment and deTection (REACT) Algorithm to approximately detect the attacked area and line failures after a cyber attack. We numerically show that it performs very well in detecting the attacked area, and detecting single, double, and triple line failures in small and large attacked areas.

KEYWORDS
Power Grids; Cyber Attacks; False Data Injection; Line Failures Detection; Graph Theory; Algorithms

1 INTRODUCTION
Due to their complexity and magnitude, modern infrastructure networks need to be monitored and controlled using computer systems. These computer systems are vulnerable to cyber attacks [1]. One of the most important infrastructure networks that is vulnerable to cyber attacks is the power grid which is monitored and controlled by the Supervisory Control And Data Acquisition (SCADA) system.

In a recent cyber attack on the Ukrainian power grid [3], the attackers stole credentials for accessing the SCADA system and used it to cause a large scale blackout affecting hundred thousands of people. In particular, they simultaneously operated several of the circuit breakers in the grid and jammed the phone lines to keep the system operators unaware of the situation [3].

Motivated by the Ukraine event, in this paper, we deploy the DC power flow model and study a model of a cyber attack on the power grid that affects both the physical infrastructure and the data at the control center. As illustrated in Fig. 1, we assume that an adversary attacks an area by: (i) disconnecting some lines within the attacked area (by remotely activating the circuit breakers), and (ii) modifying the information (phase angles of the nodes and status of the lines) received from the attacked area to mask the line failures and hide the attacked area from the control center. For the latter, we consider two types of attacks: (i) data distortion: which distorts the data by adding powerful noise to the data received from the attacked area, and (ii) data replay: which replays a locally consistent old data instead of the actual data.

We prove that the problem of finding the set of line failures given the phase angles of the nodes outside of the attacked area is strongly NP-hard, even when the attacked area is known. Hence, one cannot expect to develop a polynomial time algorithm that can exactly detect the attacked area and recover the information for all possible attack scenarios. However, we introduce the polynomial time REcurrent Attack Containment and deTection (REACT) Algorithm and numerically show that it performs very well in reasonable scenarios.

In particular, we first introduce the ATtacked Area Containment (ATAC) Module for approximately detecting the attacked area using graph theory and the algebraic properties of the DC power flow equations. We show that the ATAC Module can always provide an area containing the attacked area after a data distortion or a data replay attack. We further provide tools to improve the accuracy of the approximated attacked areas obtained by the ATAC Module under different data attack types.

Then, we introduce the randomized Line Failures Detection (LIFD) Module to detect the line failures and recover the phase angles inside the detected attacked area. The LIFD Module builds upon the methods first introduced in [30], to detect line failures using Linear Programming (LP) in more general cases. In particular, we prove that in some cases that the methods in [30] fail to detect...
line failures, the LIFD Module can successfully detect line failures in expected polynomial running time.

Finally, the REACT Algorithm combines the ATAC and LIFD Modules to provide a comprehensive algorithm for attacked area detection and information recovery following a cyber attack. We evaluate the performance of the REACT Algorithm by considering two attacked areas, one with 15 nodes and the other one with 31 nodes within the IEEE 300-bus system [2]. We show that when the attacked area is small, the REACT Algorithm performs equally well after the data distortion and the data replay attacks. In particular, it can exactly detect the attacked area in all the cases, and accurately detect single, double, and triple line failures within the attacked area in more than 80% of the cases.

When the attacked area is large, however, the REACT Algorithm’s performance is different after the data distortion and the data replay attacks. It still performs very well in detecting the attacked area after a data distortion attack and accurately detects line failures after single, double, and triple line failures in more than 60% of the cases. However, it may face difficulties providing an accurate approximation of the attacked area after a replay attack. Despite these difficulties in approximating the attacked area, it accurately detects single and double line failures in around 98% and 60% of the cases, respectively.

The main contributions of this paper are two folds: (i) analyzing the computational complexity of the attacked area detection and information recovery problem after a cyber attack on the grid, and (ii) introducing a polynomial time algorithm to address this problem and numerically evaluating its performance. To the best of our knowledge, this is the first attempt to develop recovery algorithms for the attacks in which the data from the area is modified and therefore the attacked area in unknown.

2 RELATED WORK
The vulnerability of general networks to attacks was thoroughly studied in the past (e.g., [4, 20, 28] and references therein). In particular, [10, 34] studied a problem similar to the one studied in this paper (false data injection from partial observations) in the context of communication networks.

Vulnerability of power grids to failures and attacks was extensively studied [7, 8, 13, 19, 23, 27, 29, 30]. In particular false data injection attacks on power grids and anomaly detection were studied using the DC power flows in [11, 17, 18, 21, 24, 35]. These studies focused on the observability of the failures and attacks in the grid.

The problem studied in this paper is related to the problem of line failures detection using phase angle measurements [14, 30, 32, 33, 39]. Up to two line failures detection, under the DC power flow model, was studied in [32, 33]. Since the methods developed in [32, 33] are greedy-based that need to search the entire failure space, their the running time grows exponentially as the number of failures increases. Hence, these methods cannot be generalized to detect higher order failures. Similar greedy approaches with likelihood detection functions were studied in [16, 26, 37, 38, 40] to address the PMU placement problem under the DC power flow model.

The problem of line failures detection in an internal system using the information from an external system was also studied in [39] based on the DC power flow model. The proposed algorithm works for only one and two line failures, since it depends on the sparsity of line failures. In a recent work [14], a linear multinomial regression model was proposed as a classifier for a single line failure detection using transient voltage phase angles data. Due to the time complexity of the learning process for multiple line failures, this method is impractical for detecting higher order failures.

In [30], attack scenarios similar to the one in this paper was studied. However, [30] only focused on the attacks that blocked the information from the attacked area, and therefore, the attacked area was detectable simply by checking the missing data. In this work, we build upon the results of [30] to detect line failures in more general data attack cases than the ones considered in [30]. In a recent work [31], the methods provided in [30] were extended to function under the AC power flow model.

Finally, in a recent series of works, the vulnerability of power grids to undetectable cyber-physical attacks is studied [12, 22, 36] using the DC power flows. These studies are mainly focused on designing attacks that affect the entire grid and therefore may be impossible to detect.

3 MODEL AND DEFINITIONS
3.1 DC Power Flow Model
We adopt the linearized DC power flow model, which is widely used as an approximation for the non-linear AC power flow model [6]. The notation is summarized in Table 1. In particular, we represent the power supply/demand vector \( \mathbf{p} \) as an approximation for the non-linear AC power flow model [6].

The problem studied in this paper is related to the problem of line failures detection using phase angle measurements [14, 30, 32, 33, 39]. Up to two line failures detection, under the DC power flow model, was studied in [32, 33]. Since the methods developed in [32, 33] are greedy-based that need to search the entire failure space, their the running time grows exponentially as the number of failures increases. Hence, these methods cannot be generalized to detect higher order failures. Similar greedy approaches with likelihood detection functions were studied in [16, 26, 37, 38, 40] to address the PMU placement problem under the DC power flow model.

The problem of line failures detection in an internal system using the information from an external system was also studied in [39] based on the DC power flow model. The proposed algorithm works for only one and two line failures, since it depends on the sparsity of line failures. In a recent work [14], a linear multinomial regression model was proposed as a classifier for a single line failure detection using transient voltage phase angles data. Due to the time complexity of the learning process for multiple line failures, this method is impractical for detecting higher order failures.

In [30], attack scenarios similar to the one in this paper was studied. However, [30] only focused on the attacks that blocked the information from the attacked area, and therefore, the attacked area was detectable simply by checking the missing data. In this work, we build upon the results of [30] to detect line failures in more general data attack cases than the ones considered in [30]. In a recent work [31], the methods provided in [30] were extended to function under the AC power flow model.

Finally, in a recent series of works, the vulnerability of power grids to undetectable cyber-physical attacks is studied [12, 22, 36] using the DC power flows. These studies are mainly focused on designing attacks that affect the entire grid and therefore may be impossible to detect.
Table 1: Summary of notation.

| Notation         | Description                                                                 |
|------------------|-----------------------------------------------------------------------------|
| $G = (V, E)$     | The graph representing the power grid                                       |
| $A$              | Admittance matrix of $G$                                                    |
| $\theta$        | Vector of the phase angles of the nodes in $G$                             |
| $\bar{p}$       | Vector of power supply/demand values                                         |
| $H$             | A subgraph of $G$ representing the attacked area                           |
| $F$             | Set of failed edges due to an attack                                        |
| $D$             | Incidence matrix of $G$                                                     |
| $N(i)$          | Set of neighbors of node $i$                                                |
| $N(S)$          | Set of neighbors of subgraph $S$                                            |
| $\text{int}(S)$ | Interior of the subgraph $S$                                                |
| $\partial(S)$   | Boundary of the subgraph $S$                                                |
| $\text{cl}(S)$  | Closure of the subgraph $S$                                                 |
| $\circ$         | The actual value of $\circ$                                 |
| $\circ^*$       | The observed value of $\circ$ after an attack                               |
| $\circ'$        | The complement of $\circ$                                                  |

where $A \in \mathbb{R}^{|V| \times |V|}$ is the admittance matrix of $G$, defined as:

$$a_{uv} = \begin{cases} 
0 & \text{if } u \neq v \text{ and } \{u, v\} \notin E, \\
-1/r_{uv} & \text{if } u \neq v \text{ and } \{u, v\} \in E, \\
-\sum_{w \in N(u)} a_{uw} & \text{if } u = v.
\end{cases}$$

Note that in power grids nodes can be connected by multiple edges, and therefore, if there are $k$ multiple lines between nodes $u$ and $v$, $a_{uv} = -\sum_{k=1}^{k} 1/r_{uv}$. Once $\bar{p}$ is computed, the flows, $P_{uv}$, can be obtained from (2).

Notation. Throughout this paper we use bold uppercase characters to denote matrices (e.g., $A$), italic uppercase characters to denote sets (e.g., $V$), and italic lowercase characters and overline arrow to denote column vectors (e.g., $\overline{y}$). For a matrix $Q$, $Q_l$ denotes its $i^{th}$ row, and $q_{lj}$ denotes its $(i, j)^{th}$ entry. For a column vector $\overline{y}$, $\overline{y}^T$ denote its transpose, $y_j$ denotes its $j^{th}$ entry, $||\overline{y}||_1 := \sum_{i=1}^{n} |y_i|$ is its $l_1$-norm, and $\text{supp}(\overline{y}) := \{i|y_i \neq 0\}$ is its support.

3.2 Attack Model

We study a cyber attack on the power grid that affects both its physical infrastructure and the data at its control center. We assume that an adversary attacks an area by: (i) disconnecting some lines within the attacked area (by remotely activating the circuit breakers), and (ii) modifying the information (phase angle of the nodes and status of the lines) received from the attacked area to mask the line failures and hide the attacked area from the control center. We assume that supply/demand values do not change after the attack and disconnecting lines within the attacked area does not make $G$ disconnected. However, the developed methods in this paper can also be used when these conditions do not hold, if the control center is aware of the changes in the supply/demand values after the attack and in the case of the grid separation.

Fig. 1 shows an example of such an attack on the area represented by $H = (V_H, E_H)$. Due to the attack, some edges are disconnected (we refer to these edges as failed lines) and the phase angles and the status of the lines within the attacked area are modified. We denote the set of failed lines in area $H$ by $F \subseteq E_H$. Upon failure, the failed lines are removed from the graph and the flows are redistributed according to (1)-(2). The objective is to detect the attacked area and the failed lines after the attack using the observed phase angles.

The vectors of phase angle of the nodes in $H$ and in its complement $\overline{H} = G \setminus H$ are denoted by $\theta_H$ and $\overline{\theta}_H$, respectively. We use the prime symbol (‘) to denote the actual values after an attack. For instance, $\theta'_H$, $A'_H$, and $\overline{\theta}'_H$ are used to represent the graph, the admittance matrix of the graph, and the actual phase angles after the attack. Based on our assumptions $\bar{p} = A\bar{\theta} = A'\overline{\theta}' = \bar{p}'$.

We also use $\theta^*_{H}$ to denote the observed phase angles after the attack. According to the attack model $\theta^*_{H}$ is modified and is not necessarily equal to $\overline{\theta}'_H$. We assume that the attacker performs any of the following two types of data attacks:

1. **Data distortion:** We assume $\overline{\theta}'_H = \bar{\theta}'_H + \bar{z}$ for a random vector $\bar{z}$ with an arbitrary distribution with no positive probability mass in any proper linear subspace (e.g., multivariate Gaussian distribution).

2. **Data replay:** We assume $\overline{\theta}'_H = \bar{\theta}'_H = \bar{\theta}''$ such that $\bar{\theta}''$ satisfies $A\bar{\theta}'' = \bar{p}''$ for an arbitrary power supply/demand vector $\bar{p}''$ such that $\bar{p}''_H = \bar{p}_H$. We assume that $\bar{p}''_H$ is selected generally enough and is only known to the attacker. $\bar{\theta}''$ can be considered as the vector of supply/demand values from previous hours or days.

Notice that adversarial modification of the reported phase angles in $H$ is not in the scope of this paper and is an interesting problem on its own.

Notation. Without loss of generality we assume that the indices are such that $V_H = \{1, 2, \ldots, |V_H|\}$ and $E_H = \{e_1, e_2, \ldots, e_{|E_H|}\}$. If $X, Y$ are two subgraphs of $G$, $A_{X|Y}$ and $A_{X|X} \mid Y$ both denote the submatrix of the admittance matrix of $G$ with rows from $V_X$ and columns from $V_Y$. For instance, $A$ can be written in any of the following forms,

$$A = \begin{bmatrix} A_{H|H} & A_{H|\overline{H}} \\ A_{\overline{H}|H} & A_{\overline{H}|\overline{H}} \end{bmatrix} = [A_{G|H} \ A_{G|\overline{H}}], A = \begin{bmatrix} A_{H|G} \ A_{\overline{H}|G} \end{bmatrix}.$$  

3.3 Graph Theoretical Terms

In this paper, we use some graph theoretical terms most of which are borrowed from [9].

Subgraphs: Let $X$ be a subset of the nodes of a graph $G$. $G[X]$ denotes the subgraph of $G$ induced by $X$. We denote the complement of a set $X$ by $\overline{X} = V \setminus X$.

The neighbors, interior, boundary, and closure of a subgraph $S$ are defined and denoted by $N(S) := \{i \in V \mid \exists j \in V_S : i \in N(j)\}$, $\text{int}(S) := \{i \in V_S \mid (i) \subseteq V_S\}$, $\partial(S) := \{i \in V_S \mid (i) \cap V_S \neq \emptyset\}$, and $\text{cl}(S) := V_S \cup N(S)$, respectively.

Incidence Matrix: Assign arbitrary directions to the edges of $G$. The (node-edge) incidence matrix of $G$ is denoted by $D \in \{-1, 0, 1\}^{|V| \times |E|}$ and is defined as follows,

$$d_{ij} = \begin{cases} 
0 & \text{if } e_j \text{ is not incident to node } i, \\
1 & \text{if } e_j \text{ is coming out of node } i, \\
-1 & \text{if } e_j \text{ is going into node } i.
\end{cases}$$

When we use the incidence matrix, we assume an arbitrary orientation for the edges unless we mention a specific orientation.
4 HARDNESS

Using the notation provided in the previous section, the problem considered in this paper can be stated as follows: Given $A$, $\bar{\theta}$, and $\bar{\theta}'$, detect the attacked area $H$ and the set of line failures $F$. In this section, we study the computational complexity of this and related problems. To study the computational complexity of this problem, we consider a more general case of $\theta_H$ without any assumptions on the type of the data attack.

First, we prove that the problem of finding the set of line failures ($F$) solely based on the given the phase angles of the nodes before $(\bar{\theta})$ and after the attack $(\bar{\theta}')$ is NP-hard. We prove this by reduction from the 3-proposition problem.

**Definition 4.1.** Given a set $S = \{s_1, s_2, \ldots, s_{3k}\}$ of $3k$ elements and a bound $B$, such that $\sum_{j=1}^{3k} s_j = kB$ and for $1 \leq i \leq 3k$, $B/k < s_j < B/2$, the 3-proposition problem is the problem of whether $S$ can be partitioned into $k$ disjoint sets $S_1, S_2, \ldots, S_k$ such that for $1 \leq i \leq k$, $\sum_{j \in S_i} s_j = B$ (note that each $S_i$ must therefore contain exactly 3 elements from $S$).

Lemmata 4.2 (Garey and Johnson[15]). The 3-proposition problem is strongly NP-complete.

Lemmata 4.3. Given $A$, $\bar{\theta}$, and $\bar{\theta}'$, it is strongly NP-hard to determine if there exists a set of line failures $F$ such that $A'\bar{\theta}' = A\bar{\theta}$.

**Proof.** We reduce the 3-proposition problem to this problem. Assume $S$ is a given set as described in Def. 4.1, we form a bipartite graph $G = (V, E)$ such that $V = X \cup Y$, $E = \{ (x, y) | x \in X, y \in Y \}$, $X = \{1, \ldots, k \}$, and $Y = \{k + 1, \ldots, 4k \}$. For all edges in $G$, we set the reactance values equal to 1. For each $i \in X$, we set $p_i = B$ and for each $j \in Y$ we set $p_j = -s_{j-k}$. Define the vector of phase angles $\bar{\theta}$ as follows:

$$\bar{\theta}_i = \begin{cases} 0 & i \leq k \\ -s_{i-k} & i > k. \end{cases}$$

If $A$ is the admittance matrix of $G$, it is easy to check that $A\bar{\theta} = \bar{\theta}$. Now define $\bar{\theta}'$ as follows:

$$\bar{\theta}'_i = \begin{cases} 0 & i \leq k \\ -s_{i-k} & i > k. \end{cases}$$

We prove that there exist a set of line failures $F$ such that $A'\bar{\theta}' = \bar{\theta}$ if, and only if, there exists a solution to the 3-proposition problem.

First, let's assume that there exist a solution to the 3-proposition problem such as $S_1, \ldots, S_k$. Set $E_S = \{(i, j) | s_{j-k} \in S_i \}$. We show that $F = E \setminus E_S$ implies $A'\bar{\theta}' = \bar{\theta}$. Notice that $F = E \setminus E_S$ means that $G' = (V, E \setminus E_S)$. Given the $p_i$ and the reactance values, it is easy to check that the defined $\bar{\theta}$ satisfies the DC power flow equations (1)-(2) in $G'$. Hence, $A'\bar{\theta}' = \bar{\theta}$.

Now, let's assume that there exist a set of line failures $F$ such that $A'\bar{\theta}' = \bar{\theta}$. Set $E_S = E \setminus F$ and $G' = (V, E \setminus E_S)$. Given the phase angles $\bar{\theta}'$, it is easy to see that for any $(i, j) \in E_S, p_{j-k} = s_{j-k}$. This implies that for $j \in Y$, at most one edge in $E_S$ is incident to $j$. On the other hand, using (1), for any $i \in X$, $\sum_{j \in N(i') \setminus E_S} s_{j-k} = B$ in which by $N(i')$ we mean the set of neighbors of node $i$ in $G'$. Given that each node $j \in Y$ is incident to at most one edge in $E_S$, defining $S_i = \{s_{j-k} | j \in N(i')\}$ for $1 \leq i \leq k$ gives a good solution to the 3-proposition problem.

Hence, determining if there exist a set of line failures $F$ is at least as hard as determining if the 3-proposition problem has a solution, and therefore, it is an NP-hard problem in the strong sense.

**Corollary 4.4.** Given $A$, $\bar{\theta}$, and $\bar{\theta}'$, it is strongly NP-hard to find the set of line failures $F$, even if such a set exists.

**Proof.** It is easy to see that if one can find a set of line failures $F$ with an algorithm, the output of that algorithm can be used here to verify the correctness and existence of such a set as well. Therefore, this problem is at least as hard as the existence problem.

In Corollary 4.4, we proved that given the phase angle of the nodes before and after the attack, it is NP-hard to detect the set of line failures $F$. In the following lemma, we show that even if the attack area $H$ is known (since $\bar{\theta}_H$ is not given) the problem remains NP-hard.

**Lemma 4.5.** Given $A$, $\bar{\theta}$, $H$, and $\bar{\theta}'_H$, it is strongly NP-hard to determine if there exist a set of line failures $F$ in $H$ and a vector $\bar{\theta}'_H$, such that $A'\bar{\theta}'_H = A\bar{\theta}$.

**Proof.** The idea of the proof is very similar to the proof of Lemma 4.3. Again we reduce the 3-proposition problem with a given set $S$ as described in Def. 4.1 to this problem. Consider sets $X_1 = \{1, \ldots, k\}, X_2 = \{k + 1, \ldots, 2k\}, Y_1 = \{5k + 1, \ldots, 8k\}$, and $Y_2 = \{2k + 1, \ldots, 5k\}, Y_1 = \{5k + 1, \ldots, 8k\}$. We form a bipartite graph $G = (V, E)$ such that $V = X_1 \cup X_2 \cup Y_2 \cup Y_1$ and $E = \{(i, k + 1) | 1 \leq i \leq k \} \cup \{(x, y) | x \in X_2, y \in Y_2\} \cup \{(j, j+3k) | 2k + 1 \leq j \leq 5k\}$. Notice that the defined bipartite graph here is very similar to the one defined in the proof of Lemma 4.3 except that here for each node in $X_2$ and $Y_2$ there exist a dummy node in $X_1$ and $Y_1$, accordingly, that is directly connected to its counterpart. We set $H = G[X_2 \cup Y_2]$. It is easy to see that $H$ has exactly the same topology as the graph $G$ in the proof of Lemma 4.3. Again for all edges in $G$, we set the reactance values equal to 1. For each $i \in X_2 \cup Y_2$ we set $p_i = 0$, for each $i \in X_1$, we set $p_i = B$, and for each $j \in Y_1$ we set $p_j = -s_{j-5k}$. Define the vector of phase angles $\bar{\theta}$ as follows:

$$\bar{\theta}_i = \begin{cases} B & 1 \leq i \leq k \\ 0 & k + 1 \leq i \leq 2k \\ -s_{i-2k} & 2k + 1 \leq i \leq 5k \\ -s_{i-5k} & 5k + 1 \leq i \leq 8k \end{cases}$$

If $A$ is the admittance matrix of $G$, it is easy to check that $A\bar{\theta} = \bar{\theta}$.

Now define $\bar{\theta}'$ as follows:

$$\bar{\theta}'_i = \begin{cases} B & 1 \leq i \leq k \\ 0 & k + 1 \leq i \leq 2k \\ -s_{i-2k} & 2k + 1 \leq i \leq 5k \\ -2s_{i-5k} & 5k + 1 \leq i \leq 8k \end{cases}$$

Now given $\bar{\theta}'_H$, since each node in $H$ is connected to an exactly one distinct node in $\bar{\theta}_H$, there exist a matching between the nodes in
$H$ and $\bar{H}$ that covers nodes in $H$ and therefore from [30, Corollary 2], $\bar{\theta}'_{H}$ will be determined uniquely. Hence, we can assume that $\bar{\theta}'$ is given for all the nodes. Now we prove that there exist a set of line failures $F$ in $H$ such that $A\bar{\theta}' = \bar{\theta}$ if, and only if, there exist a solution to the 3-partition problem. Given the way we build the graph $G$ and since the set of failures should be in $H$, the rest of the proof is exactly similar to the proof of Lemma 4.3.

Corollary 4.6. Given $A, \bar{\theta}, H$, and $\bar{\theta}'_{H}$, it is strongly NP-hard to find the set of line failures $F$ in $H$ if, and only if, a set of failures exists.

Finally, we prove that when the phase angles are modified ($\bar{\theta}^*$) and therefore $H$ is not known in advance, it is NP-hard to detect $H$ and $F$. We assume that the attacked area cannot contain more than half of the nodes; otherwise this problem might have many solutions.

Lemma 4.7. Given $A, \bar{\theta}$, and $\bar{\theta}^*$, it is strongly NP-hard to determine if there exists a subgraph $H_0$ with $|V_{H_0}| \leq \frac{|V|}{2}$, a set of line failures $F$ in $H_0$, and a vector $\bar{\theta}'_{H_0}$ such that $\bar{\theta} = A\bar{\theta}' = A'\bar{\theta}'_{H_0}$.

Proof. Again we reduce the 3-partition partition problem to this problem. The proof is similar to the proof of Lemma 4.5. Given an instance of a 3-partition problem, we build a graph $G$, subgraph $H$, and supply and demand vector $\bar{p}$ exactly as in the proof of Lemma 4.5. Define $\bar{\theta}'_{H} = \bar{\theta}'_{H}$ as defined in Lemma 4.5 and $\bar{\theta}'_{\bar{H}} = \bar{z}$, in which $\bar{z}$ is a random vector with arbitrary distribution with no positive probability mass in any proper linear subspace. For any $i \in X_1$, node $i$ is only connected to node $i + k$. Since $\theta_i = 0$ and $\theta_{i+k} = z_i$ for a random variable $z_i$, $\theta_i - \theta_{i+k} = 0$ almost surely. So in order for the flow equations to hold, either both $i, i + k \in H_0$ or $i + k \in H_0$. The same argument holds for any node $j \in Y_1$ and its only neighbor $j - 3k$. So in order for the problem to have a solution, $H_0$ should contain both $X_2$ and $Y_2$. On the other hand, since $|V_{H_0}| \leq \frac{|V|}{2}$, therefore $H_0 = G[X_2 \cup Y_2] = H$ is the only possible attacked area. Now since $H_0 = H$, we can assume that the attacked area is given and the rest of the proof is exactly similar to the proof of Lemma 4.5.

Corollary 4.8. Given $A, \bar{\theta}$, and $\bar{\theta}^*$, it is strongly NP-hard to find a subgraph $H$, a set of line failures $F$ in $H$, and a vector $\bar{\theta}'_{H}$ such that $A\bar{\theta} = A\bar{\theta}' = A'\bar{\theta}'_{H}$, even if such $H, F$ exist.

Corollary 4.8 indicates that it is NP-hard to detect the line failures after an attack as described in Section 3 in general cases. However, in the next sections, we provide a polynomial-time algorithm to detect the attacked area and the set of line failures $F$, and show based on simulations that it performs well in reasonable scenarios.

5 ATTACKED AREA APPROXIMATION

In this section, we provide methods to approximate the attacked area after a cyber attack as described in Subsection 3.2. In subsections 5.1 and 5.2, we first provide methods to contain the attacked area after the data distortion and replay attacks, respectively. We then combine these methods in the Attack Area Containment (ATAC) Module for containing the attacked area after both types of data attacks. Finally, given an area containing the attacked area, in subsection 5.4, we provide methods to improve the approximation of the attacked area.

5.1 Data Distortion

We first consider data distortion attacks. In particular, recall that we assume that $\bar{\theta}'_{H} = \bar{\theta}'_{H} + \bar{z}$ for a random vector $\bar{z}$ with an arbitrary distribution with no positive probability mass in any proper linear subspace. Since $\bar{\theta}'_{H}$ is the vector of the modified phase angles and there are also some line failures in $H$, it can be seen that $A\bar{\theta}' = \bar{\theta}$.

Lemma 5.1. For any $i \in \text{int}(H), A_i\bar{\theta}' = p_i$.

Proof. Since $i \in \text{int}(H)$, therefore $a_{ij} = 0$ for all $j \in V_H$. Hence, $A_i\bar{\theta}' = A_i\bar{\theta}$. On the other hand, since the attack is inside $H$, we know $A_i\bar{\theta}' = A_i\bar{\theta}$. Hence, $A_i\bar{\theta}' = A_i\bar{\theta} = p_i$.

Lemma 5.2. For any $i \in V \setminus \text{int}(H), A_i\bar{\theta}' \neq p_i$ almost surely.

Proof. For any $i \in V \setminus \text{int}(H)$, there exists a node $j \in V_H$ such that $a_{ij} \neq 0$. Now since the set of solutions $\bar{z} \rightarrow A_i\bar{z} = p_i$ is a measure zero set in $\mathbb{R}^n$ and $\theta_j$ is a random modification of $\theta_j'$, $A_i\bar{\theta}' \neq p_i$ almost surely.

Lemmas 5.1 and 5.2 indicate that given $A, \bar{\theta}$, and $\bar{\theta}'$ one can find int$(H)$ by computing $V \setminus \text{supp}(A\bar{\theta}' - \bar{\theta})$.

Corollary 5.3. int$(H) = V \setminus \text{supp}(A\bar{\theta}' - \bar{\theta})$, almost surely.

Define $S_0 := G[\text{supp}(A\bar{\theta}' - \bar{\theta})]$. We know from Corollary 5.3 that int$(H) = V_{S_0}$ and from Lemma 5.2 that $V_H \subset V_{S_0}$. Therefore, $S_0$ clearly contains $H$. The following lemma demonstrates that int$(S_0)$ is a better approximation for $V_H$. We use this lemma in Subsection 5.4 to improve the approximation of the attacked area.

Lemma 5.4. $V_H \subset \text{int}(S_0)$, almost surely.

Proof. Assume not. Then there exists a node $i \in V_H$ such that $N(i) \cap V_{S_0} \neq \emptyset$. Assume $j \in N(i) \cap V_{S_0} \neq \emptyset$, then with a similar argument as in the proof of Lemma 5.2, one can show that $A_i\bar{\theta}' \neq p_i$ almost surely, which contradicts with $j \notin V_{S_0}$. Hence, $N(i) \cap V_{S_0} = \emptyset$ and $V_H \subset \text{int}(S_0)$.

5.2 Data Replay

In this subsection, we consider data replay attacks. Recall that we assume $\bar{\theta}'_{H} = \bar{\theta}'_{H}$ such that $\bar{\theta}'$ satisfies $A\bar{\theta}' = \bar{\theta}'$. The power supply/demand vector $\bar{p}'$ is arbitrarily selected such that $\bar{p}' = \bar{p}$, and $\bar{p}' = \bar{p}'$ is selected generally enough.

The data replay attacks are harder to detect since the data seems to be correct locally. Again, one can easily see that $A\bar{\theta}' \neq \bar{\theta}$, but here unlike the data distortion case, not all the nodes in $H$ can be detected by checking $A_i\bar{\theta}' \neq p_i$. The following lemma shows why attacked area containment is more difficult after a data replay attack.

Lemma 5.5. For any $i \in \text{int}(H) \cup \text{int}(\bar{H}), A_i\bar{\theta}' = p_i$. 
Proof. Similar to the proof of Lemma 5.1, it is easy to show that for any \( i \in \text{int}(H) \), \( A_i \vec{\theta}^* = p_i \). The only new part is to show the same for nodes in \( \text{int}(\bar{H}) \). So assume \( i \in \text{int}(\bar{H}) \), following the definition of the interior, it can be verified that \( A_i \vec{\theta}^* = A_i|\bar{H} \vec{\theta}^*_\bar{H} \). On the other hand, since \( \vec{\theta}_H^* = \vec{\theta}'_H^* \) and \( \bar{p}_j'' = \bar{p}_j \), we can verify that \( A_i|\bar{H} \vec{\theta}_H'' = A_i|\bar{H} \bar{p}_i' = p_i \). Hence, for all \( i \in \text{int}(H) \) also \( A_i \vec{\theta}^* = p_i \). \( \square \)

**Lemma 5.6.** For any \( i \in \partial(H) \cup \partial(\bar{H}) \), \( A_i \vec{\theta}^* \neq p_i \), almost surely.

**Proof.** The proof of this lemma is similar to the proof of Lemma 5.2. For any \( i \in \partial(\bar{H}) \), there exists a node \( j \in V_H \) such that \( a_{ij} \neq 0 \). Now since the set of solutions \( \vec{x} \) to \( A_i \vec{x} = p_i \) is a measure zero set in \( \mathbb{R}^n \) and \( \vec{\theta}''_j = \vec{\theta}'_j^* \) for a generally enough selected vector \( \bar{p}_j'' \), \( A_i \vec{\theta}^* \neq p_i \) almost surely. A similar argument holds for \( i \in \partial(H) \). \( \square \)

**Corollary 5.7.** \( \text{supp}(\vec{A} \vec{\theta}^* - \bar{p}) = \partial(H) \cup \partial(\bar{H}) \), almost surely.

From comparing Corollaries 5.3 and 5.7, one can see that in the replay attack case, \( \bar{S}_0 = \{ \text{supp}(\vec{A} \vec{\theta}^* - \bar{p}) \} \) does not contain the attacked area \( H \) anymore. In the following lemma, we show how one can still contain the attacked area in this case.

**Lemma 5.8.** If \( C_1, C_2, \ldots, C_k \) are the connected components of \( G \backslash \text{supp} \), then these connected components can be divided into disjoint sets \( \{I_1, i_2, \ldots, i_l\} \) such that \( \text{int}(\bar{H}) \) is the connected components obtained in \( C_i \cup C_j \cup \ldots \cup C_{l-1} \) and \( \text{int}(\bar{H}) \) is the connected components obtained in \( C_j \cup C_{l-1} \cup \ldots \cup C_{l-1} \).

**Proof.** It is a direct result of Corollary 5.7. \( \square \)

**Lemma 5.9.** For two connected components \( C_i \) and \( C_j \) of \( \text{G} \backslash \text{supp} \), if \( N(C_i) \cap N(C_j) \neq \emptyset \), then either \( C_i \cup C_j \subseteq \text{int}(H) \) or \( C_i \cup C_j \subseteq \text{int}(\bar{H}) \).

**Proof.** From Lemma 5.8, for any \( i \), either \( C_i \subseteq \text{int}(\bar{H}) \) or \( C_i \subseteq \text{int}(H) \). If \( C_i \subseteq \text{int}(\bar{H}) \) then \( N(C_i) \subseteq \partial(\bar{H}) \), and if \( C_i \subseteq \text{int}(H) \) then \( N(C_i) \subseteq \partial(H) \). Hence, since \( \partial(\bar{H}) \cap \partial(H) = \emptyset \), if \( N(C_i) \cap N(C_j) \neq \emptyset \), then either \( C_i \cup C_j \subseteq \text{int}(H) \) or \( C_i \cup C_j \subseteq \text{int}(\bar{H}) \). \( \square \)

Following Lemma 5.9, one can see that connected components \( C_1, C_2, \ldots, C_k \) can be combined into disjoint subgraphs \( G_1, G_2, \ldots, G_t \) such that for any two of these subgraphs such as \( G_i \) and \( G_j \), \( N(G_i) \cap N(G_j) = \emptyset \). Moreover for any \( i \), either \( G_i \subseteq \text{int}(H) \) or \( G_i \subseteq \text{int}(\bar{H}) \). In the following lemma, we use this fact to contain the attacked area.

**Lemma 5.10.** There exists \( 1 \leq i \leq t \), such that \( H \subseteq G\backslash G_i \). Moreover, \( H \subseteq \text{int}(G\backslash G_i) \).

**Proof.** The first part of the proof is the direct result of Lemmas 5.8 and 5.9. To prove the second part, notice that for any \( i \), \( S_0 \subseteq G\backslash G_i \). Therefore, for any \( i \), \( \partial(\bar{H}) \subseteq G\backslash G_i \). Hence, if \( H \subseteq G\backslash G_i \), since \( \partial(\bar{H}) \subseteq G\backslash G_i \), one can verify that \( H \subseteq \text{int}(G\backslash G_i) \). \( \square \)

Lemma 5.10 demonstrates that at least one of \( G\backslash G_i \) contains the attacked area. Hence, one can use this fact to contain the attacked area after a data replay attack.

**5.3 The ATAC Module**

Using the results in the previous subsections, here we introduce the ATAC module. As shown in Fig. 2, there are scenarios for which the data attack type cannot be recognized by simply looking at \( S_0 \). Hence, the ATAC module does not return a single subgraph containing the attacked area but a series of possible subgraphs. In Sections 6 and 7, we show that by defining the confidence of the solution, an algorithm can go over all of these subgraphs until it detects the attacked area and the set of line failures with high confidence.

The steps of the ATAC module are summarized in Module 1. As can be seen, \( S_0 \) is the first possible subgraph returned by the ATAC module, which is for the case when there is a data distortion attack. Then based on Lemma 5.10, \( S_1 := G\backslash G_1, S_2 := G\backslash G_2, \ldots, S_t := G\backslash G_t \) are other possible areas containing the attacked area, if there is a replay attack. Notice that since \( t < |V| \), therefore the ATAC module is a polynomial time algorithm.

**5.4 Improving Attacked Area Approximation**

Assume that from the subgraphs returned by the ATAC module, \( S^* \) is one of them that contains the attacked area \( H \). Following Lemma 5.4 and Lemma 5.10, \( S_2 := G\backslash \text{int}(S^*) \) is a better approximation for the attacked area \( H \). In order to find a more accurate approximation for \( H \), we provide the following lemma which is similar to [30, Lemma 1].

**Lemma 5.11.** For a subgraph \( S \), if \( V_H \subseteq V_S \), then:

\[
A_{S|G}(\vec{\theta} - \bar{p}) = 0.
\] (4)

**Proof.** Since all the line failures are inside \( H \), and also \( V_H \subseteq V_S \), therefore it can be seen that \( A_{S|G} = A'_{S|G} \). On the other hand,
and yellow nodes represent the nodes in $\partial H$ set and $\overline{\partial H}$ set, in which $\overline{\partial H}$ includes all $\partial H$ and unknown as follows:

$$H = S \cup \overline{\partial H}.$$  \hfill ($\star$)

Since $\overline{\partial H} \subseteq V_S$, therefore $\overline{\partial H} = \overline{\partial H}^*$. Hence, the only unknown variable in the above system $\delta_S$ is a solution to the following linear system:

$$\begin{align*}
A_{\overline{S}|S} \delta_S - \delta_S^* &= \bar{p}_S^	op, \\
A_{\overline{S}|S} \delta_S - \delta_S^* &= \bar{p}_S.
\end{align*}$$

(5)

Proof. If there is a matching between the nodes inside and outside of $H$ that covers all the nodes in $\partial(S)$, one can prove that $A_{\overline{S}|\partial(S)}$ has linearly independent columns, almost surely (see [30, Corollary 2]). Moreover, it is easy to see that $A_{\overline{S}|\partial(S)} = 0$. Hence, if $\bar{y}$ is a solution to (5), $y_{|\partial(H)} = \delta_H^*$. Now since $V_S \setminus V_H \subseteq \partial(S)$, for any $i \in V_S \setminus V_H$, $y_i = \delta_H^*$. On the other hand, since $\delta_H^*$ is selected generally enough, one can verify that for any $i \in H$, $y_i \neq \delta_i^*$, almost surely. Therefore, $G[\text{supp}(\bar{y} - \delta_S^*)] = H$, almost surely.

\hfill $\Box$

6 LINE FAILURE DETECTION

In the previous section, we provided methods to find a good approximation $S$ for the attacked area $H$. In this section, we provide a method to detect line failures inside $S$. For this reason, we use and build on the idea introduced in [30]. It was proved in [30] that if the attacked area $H$ is known, then there always exists feasible vectors $x \in \mathbb{R}^{|E_H|}$ and $\bar{y} \in \mathbb{R}^{|V_H|}$ satisfying the conditions of the following optimization problem such that supp($\bar{y}$) = $F$ and $\bar{y} = \delta_H^*$:

$$\begin{align*}
\min \|x\|_1 \text{ s.t.} \\
A_{H|H}[\delta_H - y] + A_{H|H}[\delta_H - \delta_H^*] &= D_H x, \\
A_{H|H}[\delta_H - y] + A_{H|H}[\delta_H - \delta_H^*] &= 0.
\end{align*}$$

(6)

Notice that the optimization problem (6) can be solved efficiently using Linear Program (LP). It is proved in [30] that under some conditions on $H$ and the set of line failures $F$, the solution to (6) is unique, therefore the relaxation is exact and the set of line failures can be detected by solving (6). In particular, it is proved in [30] that if $H$ is acyclic and there is a matching between the nodes in $H$ and $\overline{\partial H}$ that covers $H$, the solution to (6) is unique for any set of line failures.

Since the conditions on $H$ and $F$ as described in [30] may not always hold for the exactness of the line failures detection using (6), it cannot be used in general cases to detect line failures. To address this issue, here, we introduce a randomized version of (6).

Assume that $W = \mathcal{W}[\mathcal{W}]$ is a diagonal matrix. We show that the solution to the following optimization problem can detect line failures in $S$ accurately for a "good" matrix $W$:

$$\begin{align*}
\min \|x\|_1 \text{ s.t.} \\
A_{\overline{S}|S}[\delta_S - \bar{y}] + A_{\overline{S}|S}[\delta_S - \delta_S^*] &= D_S x, \\
A_{\overline{S}|S}[\delta_S - \bar{y}] + A_{\overline{S}|S}[\delta_S - \delta_S^*] &= 0.
\end{align*}$$

(7)

The idea behind optimizing the weighted norm-1 of vector $x$ is to be able to detect the line failures when the solution to (6) does not detect the correct set of line failures but a small disturbance results in the correct detection.

Before we demonstrate the effectiveness of the optimization (7) in detecting line failures, we provide a metric for measuring the confidence of a solution. In a subgraph $S$, assume $F^\perp = \text{supp}(\bar{x})$ and $\delta_S^\perp = \bar{y}$ are the set of detected line failures and the recovered phase angles using the solution to (7). Also, assume that $A^\perp = A_{G|S}[\delta_S]^{\perp}$ is the admittance matrix after removing the lines in $F^\perp$ and define $\delta_S^\perp := A_{G|S}[\delta_S]^{\perp}$, $\bar{y}^\perp = A_{G|S}[\delta_S]^{\perp}$. Notice that $\bar{x}$ and $\bar{y}$ satisfying (7) does not match. Therefore, $G[\text{supp}(\bar{y} - \delta_S^*)] = H$, almost surely.

Figure 3: $H$ is an induced subgraph of $G$ that represents the attacked area with data distortion attack. The red, orange, and yellow nodes represent the nodes in $S_b, S_a \setminus S_b$, and $S^* \setminus S_a$, respectively.
Module 2: Line Failures Detection (LIFD)

**Input:** G, A, Ŝ, and Ŝ*

1. Compute \( \hat{p} = \theta^T A \theta \)
2. Compute a solution \( \hat{x}, \hat{y} \) to (7) for \( W = I \)
3. Set \( F^i = \text{supp}(\hat{x}) \) and \( \hat{\delta}^i_S = \hat{y} \)
4. while \( c(F^i, \hat{\delta}^i_S) < 99.99\% \) & counter \( T \) do
   5. counter \( \leftarrow \) counter + 1
   6. Draw random numbers \( w_1, w_2, \ldots, w_{|V_S|} \) from an exponential distribution with rate \( \lambda = 1 \)
   7. Compute a solution \( \hat{x}, \hat{y} \) to (7) for \( W = \text{diag}(w_1, w_2, \ldots, w_{|V_S|}) \)
   8. Set \( F^i = \text{supp}(\hat{x}) \) and \( \hat{\delta}^i_S = \hat{y} \)
   9. if \( c(F^i, \hat{\delta}^i_S) > 99.99\% \) then
      10. return \( F^i, \hat{\delta}^i_S \) with maximum \( c(F^i, \hat{\delta}^i_S) \) in all iterations
      11. else
      12. return \( F^i, \hat{\delta}^i_S \) with maximum \( c(F^i, \hat{\delta}^i_S) \) in all iterations

not necessarily imply \( \hat{p}^i \sim \hat{p} \). Hence, one can use this difference to compute the confidence of a solution as follows.

**Definition 6.1.** The confidence of the solution is denoted by \( c(F^i, \hat{\delta}^i_S) \) and defined as:

\[
c(F^i, \hat{\delta}^i_S) := (1 - \frac{\|\hat{p} - \hat{\delta}\|_2}{\|\hat{\delta}\|_2})^2 \times 100,
\]

in which \( (z)^+ := \max(0, z) \).

The confidence of the solution along with a random selection of the weight matrix \( W \) in (7) can be used to detect line failures that cannot be detected using (6). The idea is to repeatedly solve (7) using a random weight matrix until the confidence of the solution for \( F^i = \text{supp}(\hat{x}) \) and \( \hat{\delta}^i_S = \hat{y} \) is 100\% or reach a maximum number of iterations \( T \). Here, we consider the case when the diagonal entries of matrix \( W \) are randomly selected from an exponential distribution. This approach is summarized in Module 2 as the Line Failures Detection (LIFD) Module.

Through the rest of this section, we demonstrate why the LIFD Module is effective and when the number of iterations \( T \) is enough to be polynomial in terms of the input size to make sure that it finds the line failures accurately.

**Lemma 6.2.** Assume \( w_1, w_2, \ldots, w_m \) are i.i.d. exponential random variables, then for \( 1 \leq k \leq m - 1 \):

\[
Pr\left( \sum_{i=k+1}^{m} w_i < \sum_{j=k}^{m-1} \frac{(m-1)}{2m-1} \right) = \frac{\sum_{j=k}^{m-1} (m-1)}{2m-1}.
\]

**Proof.** See Section 10.

**Corollary 6.3.** Assume \( w_1, w_2, \ldots, w_m \) are i.i.d. exponential random variables, then for \( k \leq m/2 + \Theta(\sqrt{m}) \):

\[
Pr\left( \sum_{i=1}^{k} w_i < \sum_{i=k+1}^{m} w_i \right) = \Omega\left( \frac{1}{\sqrt{m}} \right).
\]

**Proof.** See Section 10.

**Lemma 6.4.** If \( S = H \), \( H \) is a cycle with \( m \) nodes and edges, and there is a matching between the nodes inside and outside of \( H \) that covers all the inside nodes, then any set of line failures of size \( k \) can be found by the LIFD Module for expectedly \( T = 2^{m-1}/(\sum_{j=k}^{m-1} (m-1)) \).

Moreover, if \( k \leq m/2 + \Theta(\sqrt{m}) \), then LIFD Module can detect line failures for \( T = O(\sqrt{m}) \).

**Proof.** First, one can see that if \( S = H \), and there is a matching between the nodes inside and outside of \( H \) that covers all the inside nodes, then \( A_{S|S} = A_{H|H} \) has uniquely independent columns, almost surely [30, Corollary 2]. Hence, the solution \( \hat{y} \) to (7) is unique and \( \hat{y} = \hat{\delta}^i_S \). Therefore, we can assume that \( \hat{\delta}^i \) is given. Without loss of generality assume that \( F = \{e_1, \ldots, e_{k} \} \). We prove that the solution \( \hat{x} \) to (7) is unique and \( \text{supp}(\hat{x}) = F \), if \( \sum_{i=1}^{k} w_i < \sum_{i=k+1}^{m} w_i \), in which \( W = \text{diag}(w_1, \ldots, w_m) \).

Without loss of generality, assume that \( D_H \) is the incidence matrix of \( H \) when lines of \( H \) are oriented clockwise. Since \( H \) is connected, it is known that \( \text{rank}(D_H) = m - 1 \) [5, Theorem 2.2]. Therefore, \( \text{dim}(\text{Null}(D_H)) = 1 \). Suppose \( \bar{z} \in \mathbb{R}^{|E_H|} \) is the all one vector. It can be verified that \( \text{Null}(D) = 0 \). Since \( \text{dim}(\text{Null}(D_H)) = 1 \), \( \bar{z} \) forms a basis for the null space of \( D \). Now suppose \( \hat{x} \) is a solution to \( A_{H|G}(\hat{\delta}^i - \hat{\delta}^i) = D_H \hat{x} \) such that \( \text{supp}(\hat{x}) = F \) (from [30, Lemma 2]), we know that such a solution exists. Since \( \bar{z} \) forms a basis for \( \text{Null}(D) \), all other solutions of \( A_{H|G}(\hat{\delta}^i - \hat{\delta}^i) = D_H \hat{x} \) can be written in the form of \( \hat{x} + c \bar{z} \). We want to prove that if \( \sum_{i=1}^{k} w_i < \sum_{i=k+1}^{m} w_i \), then for any \( c \in \mathbb{R} \), \( ||W(\hat{x} + c \bar{z})||_2 < ||W(\hat{x})||_2 \).

Since \( \text{supp}(\hat{x}) = F, x_1, x_2, \ldots, x_k \) are the only nonzero elements of \( x \). Moreover \( W_d := \sum_{i=k+1}^{m} w_i - \sum_{i=1}^{k} w_i > 0 \). Hence,

\[
||W(\hat{x} + c \bar{z})||_2 = \sum_{i=1}^{k} w_i |x_i|^2 - c + |c| \sum_{i=k+1}^{m} w_i |x_i|
\]

\[
\geq \sum_{i=1}^{k} w_i |x_i|^2 + |c||W_d| > \sum_{i=1}^{k} w_i |x_i|^2 = ||W\hat{x}^\dagger||_2.
\]

Therefore, the solution \( \hat{x} \) to (7) is unique and \( \text{supp}(\hat{x}) = F \), if \( \sum_{i=1}^{k} w_i < \sum_{i=k+1}^{m} w_i \). One the other hand, Lemma 6.2, \( Pr\left( \sum_{i=1}^{k} w_i < \sum_{i=k+1}^{m} w_i \right) = \frac{\sum_{i=k+1}^{m} (m-1)}{2m-1} \). Hence, expectedly \( \frac{2^{m-1}}{2m-1} \) number of iterations \( T \) should be enough to satisfy this inequality. Corollary 6.3 also gives the expected number of iterations needed when \( k \leq m/2 + \Theta(\sqrt{m}) \).
7 REACT ALGORITHM

In this section, we present the REcurrent Attack Containment and deTection (REACT) Algorithm based on the results presented in the previous sections. The steps of the REACT Algorithm are summarized in Algorithm 1.

Algorithm 1: REcurrent Attack Containment and deTection (REACT)

Input: G, A, \( \hat{b} \), \( \hat{\theta}^* \) and T

1. Compute \( \bar{p} = A\hat{b} \).
2. Obtain \( S_0, S_1, \ldots, S_t \) using the ATAC Module.
3. for \( i = 1 \) to \( t \) do
   4. Compute \( S_a = G[\text{int}(S_i)] \).
   5. if (5) is feasible for \( S_a \) then
      6. Find a solution \( \bar{y} \) to (5) for \( S = S_a \).
   7. else continue.
   9. Compute \( S_b = G[\text{supp}(\bar{y} - \hat{\theta}^+_H)] \).
10. Compute a solution \( \bar{x}, \bar{y} \) to (7) for \( W = I \).
11. if \( c(F^1, \hat{\theta}^+_S) < 99.99\% \) then
12. Obtain \( F^1, \theta^+_S \) from module LIFD for inputs \( S \) and \( T \).
13. if \( c(F^1, \theta^+_S) > 99.99\% \) then
14. return \( H = \text{supp}(\hat{\theta}^+_S - \hat{\theta}^+_H) \) as the detected attacked area and \( F^1, \theta^+_H \) as the detected line failures and recovered phase angle of the nodes inside \( H \).
15. return \( S \) and \( F^1, \theta^+_H \) with maximum \( c(F^1, \theta^+_H) \) in all iterations.

8 NUMERICAL RESULTS

In this section, we evaluate the performance of the REACT Algorithm in detecting the attacked area and recovering the information after a cyber attack as described in Section 3.2. We consider two attacked areas \( H_1 \) and \( H_2 \) within the IEEE 300-bus system [2] as depicted in Fig. 4. \( H_1 \) has 15 nodes and 16 edges, and \( H_2 \) which contains \( H_1 \), has 31 nodes and 41 edges. It can be verified that none of these two subgraphs are acyclic and there is no matching between the nodes inside and outside of these two subgraphs that covers their insides nodes. Hence, the methods provided in [30] cannot recover the information inside these areas even when the attacked areas are known in advance.

For the physical part of the attack, we consider all single line failures, and 100 samples of all double and triple line failures within \( H_1 \) and \( H_2 \). Figs. 6 and 7 illustrate the REACT Algorithm’s performance after these attacks. In the Algorithm, we set \( T = 20 \) so that the while loop in the LIFD Module runs only for 20 iterations.

Fig. 6 shows the performance of the REACT Algorithm in detecting the attacked area and recovering the information after data distortion and data replay attacks on the attacked area \( H_1 \) accompanied by single, double, and triple line failures. As can be seen in Fig. 6(a), the REACT Algorithm can exactly detect the attacked area after all attack scenarios under both the distortion attack and the replay attack. Hence, the performance of the REACT Algorithm is almost the same in detecting line failures and recovering the phase angles after both data attack scenarios.

Finally, Fig. 6(f) shows the average percentage error in the recovered phase angles. It can be seen that the phase angles inside the attacked area can be recovered with less than 3%, 5%, and 7% error after the single, double, and triple line failures, respectively.

As we observe in Fig. 6, when the attacked area is relatively small, the REACT Algorithm performs very similarly after the two types of data attack. However, as it can be clearly seen in Fig. 7, it is not the case as the attacked area becomes larger. Before we analyze the results provided in Fig. 7, in order to better show the difficulty of detecting the attacked area after a data replay attack, we depicted in Fig. 5 one of the analyzed attacked scenarios in Fig. 7. As can be seen in Fig. 5(a), the REACT Algorithm can exactly detect the attacked area after a data replay attack on the same area with the
Figure 4: The two attacked areas in the IEEE 300-bus systems that are used in simulations. The red octagon nodes are the nodes in $H_1$ and $H_2$, and the orange square nodes are the nodes that are only in $H_2$.

Figure 5: The difference in difficulty of detecting the attacked area after a data distortion attack and a data replay attack on the attacked area $H_2$ accompanied by a triple line failure within $H_2$. The yellow filled nodes represent the nodes in the detected attacked area by the REACT Algorithm, the nodes with a thick red border represent the nodes in $H_2$ that are actually attacked, and blue empty nodes represent the rest of the nodes.

same set of line failures. Recall from Subsection 5.2 that the main reason for this is the difficulty of distinguishing between the nodes in $\text{int}(H)$ and $\text{int}(\bar{H})$.

Fig. 7(a) shows the extra nodes that are incorrectly detected by the REACT Algorithm as part of the attacked area. As can be seen, in the case of the data distortion attack, the number of line failures do not significantly affect the performance of the REACT Algorithm. However, in the case of the replay attack, as the number of line failures within the attacked area increases, the REACT Algorithm provides less accurate approximation of the attacked area.

Despite its difficulty in detecting the attacked area after a data replay attack, Figs. 7(b) and 7(c) demonstrate that the REACT Algorithm detects the line failures relatively accurately. For example, the REACT Algorithm accurately detects the single and double line failures in 95% and 65% of the cases, respectively.

As can be seen in Fig. 7(d), the running time of the REACT Algorithm increases as the size of the attacked area increases. However, it still detects line failures much faster than existing brute force methods [32, 33, 38–40] (to the best of our knowledge there are no methods for detecting the attacked area).
Figure 6: The REACT Algorithm’s performance in detecting the attacked area and recovering the information after data distortion and replay attacks on the attacked area $H_1$ accompanied by single, double, and triple line failures. (a) Average number of extra nodes detected as attacked in detecting the attacked area, (b) average number of false positives and negatives in detecting line failures, (c) percentage of the cases with exact line failures detection, (d) running time of the algorithm, (e) average confidence of the solutions, and (f) average error in recovered phase angles.

Similar to the previous attack scenario, one can see in Fig. 7(e) that the confidence of the solutions obtained by the REACT Algorithm are very high. It means that in these attack scenarios, many good solutions exist near the optimal solution. This demonstrates another difficulty of dealing with recovery of information after a cyber attack on the power grid.

Finally, Fig. 7(f) indicates that the REACT Algorithm performs very well in recovering the phase angles in this case as well. As can be seen, for both the data distortion and the data replay attacks accompanied by single, double, and triple line failures, the REACT Algorithm recovers the phase angles with less than 5% error.

Overall, the simulation results in this section demonstrate that the REACT Algorithm performs very well in detecting the attacked area and the line failures when the attacked area is relatively small.

Figure 7: The REACT Algorithm’s performance in detecting the attacked area and recovering the information after data distortion and replay attacks on the attacked area $H_2$ accompanied by single, double, and triple line failures. (a) Average number of extra nodes detected as attacked in detecting the attacked area, (b) average number of false positives and negatives in detecting line failures, (c) percentage of the cases with exact line failures detection, (d) running time of the algorithm, (e) average confidence of the solutions, and (f) average error in recovered phase angles.

As the attacked area becomes larger, the Algorithm still performs very well in detecting the attacked area after a distortion data attack. However, it may face difficulties providing an accurate approximation of the attacked area after a replay attack. Despite this, in both data attack scenarios, it detects line failures relatively well. One of the important observations in this section is that the LIFD module outperforms the methods provided in [30] for detecting line failures with an slight increase in the running time, since it needs to find a solution to (7) several times instead of once. The results in this section clearly demonstrate that in the attacked areas $H_1$ and $H_2$ that do not have the conditions provided [30], the LIFD Module can still detect the line failures accurately with less than 20 iterations. In most of these cases, the LIFD Module detects the line failures within much fewer number of iterations.
9 CONCLUSION

In this paper, we considered a model for cyber attacks on power grids focusing on both data distortion and data reply attacks. We proved that the problem of detecting the line failures after such an attack is NP-hard in general and even when the attacked area is known. However, using the algebraic properties of the DC power flows, we developed the polynomial time REACT Algorithm for approximating the attacked area and detecting the line failures after a cyber attack on the grid. We numerically showed that the REACT Algorithm obtains accurate results when there are few number of line failures and the attacked area is small. We showed that as the attacked area becomes larger and the number of line failures increases, the REACT Algorithm faces some difficulties but still can approximate the attacked area and detect line failures with few false negatives and positives.

We analytically and numerically showed that the data replay attacks are harder to deal with than the data distortion attacks. It is possible for an adversary to devise more sophisticated attacks to further obscure the system’s state. Studying more sophisticated attacks and improving the methods to protect the grid against such attacks is part of our future work. Moreover, since the DC power flows only provide an approximation for the more accurate AC power flows, we plan to extend our methods to function under the AC power flow model as well.

10 APPENDIX: OMITTED PROOFS

Proof of Lemma 6.2. Define \( s_k := \sum_{i=1}^{k} w_i \). It is known that

\[
\begin{align*}
    f_{s_k}(x) &= \frac{e^{-\lambda x} (\lambda x)^{k-1}}{(k-1)!}.
\end{align*}
\]

Now since \( w_i \)'s are i.i.d. random variables, \( \sum_{i=1}^{m} w_i \sim s_{m-k} \). Therefore, all we need to compute is \( \text{Pr}(s_k < s_{m-k}) \).

\[
\begin{align*}
    \text{Pr}(s_k < s_{m-k}) &= \text{Pr}(s_{m-k} - s_k = a) \text{ da} \\
    &= \int_{0}^{\infty} \int_{0}^{\infty} \text{Pr}(s_{m-k} = y + a) \text{ dy da} \\
    &= \int_{0}^{\infty} \int_{0}^{\infty} e^{-\lambda y} (\lambda y)^{m-k-1} \frac{(\lambda(y+a))^k}{(k-1)!} \text{ dy da} \\
    &= \int_{0}^{\infty} \frac{e^{-\lambda a}(y+a)^k}{(m-k)!} \int_{0}^{\lambda y} e^{-\lambda z} \text{ dz da}.
\end{align*}
\]

On the other hand, by defining \( z := \lambda y + a \), we have:

\[
\int_{0}^{\infty} e^{-\lambda a}(y+a)^{m-k-1} \text{ dy da} = \frac{e^{\lambda y}}{\lambda m-k} \int_{0}^{\lambda y} e^{-\lambda z} \text{ dz}. 
\]

Define \( T(n+1) := \int_{\lambda y}^{\infty} e^{-\lambda z} z^n \text{ dz} \). Using partial integration:

\[
T(n+1) = -e^{-\lambda y} z^{n+1} \bigg|_{\lambda y}^{\infty} + \int_{\lambda y}^{\infty} n z^n e^{-\lambda z} \text{ dz} \\
= e^{-\lambda y} (\lambda y)^n + n T(n) = n! e^{-\lambda y} \sum_{i=0}^{n} \frac{(\lambda y)^i}{i!}.
\]

Using equation above in (9) results in:

\[
\begin{align*}
    \text{Pr}(s_k < s_{m-k}) &= \frac{\lambda^k}{(k-1)!} \int_{0}^{\infty} e^{-\lambda y} \frac{1}{(m-k)!} \int_{0}^{\infty} e^{-\lambda z} \text{ dz dy} \\
    &= \frac{\lambda^k}{(k-1)!} \int_{0}^{\infty} e^{-\lambda y} \frac{1}{(m-k)!} \sum_{i=0}^{m-k-1} \frac{(\lambda y)^i}{i!} \text{ dy} \\
    &= \frac{\lambda^k}{(k-1)!} \sum_{i=0}^{m-k-1} \int_{0}^{\infty} e^{-\lambda y} \frac{1}{(m-k)!} \sum_{i=0}^{m-k-1} \frac{(\lambda y)^i}{i!} \text{ dy}.
\end{align*}
\]

By defining \( x := 2\lambda y \) and using Gamma function:

\[
\begin{align*}
    \text{Pr}(s_k < s_{m-k}) &= \frac{\lambda^k}{(k-1)!} \sum_{i=0}^{m-k-1} \int_{0}^{\infty} e^{-x} x^{i+1} \text{ dx} \\
    &= \frac{\lambda^k}{(k-1)!} \sum_{i=0}^{m-k-1} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx} \\
    &= \frac{\lambda^k}{(k-1)!} \sum_{i=0}^{m-k-1} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx} \\
    &= \frac{\lambda^k}{(k-1)!} \sum_{i=0}^{m-k-1} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx} \\
    &= \frac{\lambda^k}{(k-1)!} \sum_{i=0}^{m-k-1} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx}.
\end{align*}
\]

Now notice that \( \sum_{i=0}^{m-k-1} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx} \) is equal to the total number of subsets of \( \{1, \ldots, m-1\} \) with at least \( k \) elements. The reason is that this summation is equal to the total number of subsets that contain \( k+i \) and exactly \( k-1 \) elements from \( \{1, 2, \ldots, k+i-1\} \). It is easy to verify that by summing this up on \( i \), we count all the subsets of \( \{1, \ldots, m-1\} \) with at least \( k \) elements. On the other hand, we can count the total number of subsets of \( \{1, \ldots, m-1\} \) with at least \( k \) elements using the complement rule. The total number of subsets with at least \( k \) elements is equal to the total number of subsets minus number of subsets of size \( 0, 1, \ldots, k-1 \). Hence

\[
\sum_{i=0}^{m-k-1} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx} = \sum_{i=0}^{m-1-k} \frac{\lambda^{k-i}}{(i+1)!} \int_{0}^{\infty} e^{-x} \frac{1}{(k+i-1)!} \text{ dx}.
\]

Now using the equation above in (10) and using the equality \( 2^{m-1} = \sum_{j=0}^{m-1} \binom{m-1}{j} \), proves the lemma.

Proof of Corollary 6.3. It is easy to see that if \( k \leq (m-1)/2 \), then \( \sum_{j=k}^{m-1} \binom{m-1}{j} \geq 2^{m-2} \). Therefore from Lemma 6.2, \( \text{Pr}(s_k < s_{m-k}) \geq 2^{m-2} \) and there is nothing left to prove. So assume \( k = m/2 + O(\sqrt{m}) \). It is proved in [25, Lemma 10.8] that for any \( 1/2 < \alpha < 1 \),

\[
\frac{2^{\alpha n}}{8n a(1-a)} \leq \sum_{j=0}^{n} \binom{n}{j}.
\]

in which \( H(\alpha) = -\alpha \log_2(\alpha) - (1-\alpha) \log_2(1-\alpha) \) is the entropy function. Now to prove Corollary 6.3, select \( n = m-1 \), and \( \alpha = 1/2 + \epsilon \) for \( \epsilon = \Theta(1/\sqrt{n}) \). First notice that one can show that the Taylor expansion of the entropy function around 1/2 can be computed as:

\[
H(\alpha) = 1 - \frac{1}{2} \log_{2} 2 \sum_{i=1}^{\infty} \frac{(1-2\alpha)^i}{i(i-1)}.
\]
Using approximation above, it is easy to see that $H(a) ≈ 1−\Theta(\epsilon^2) = 1−\Theta(1/n)$. Hence, $2^{n H(a)} \approx 2^{n−\Theta(1)}$. On the other hand,

$$\sqrt{8n(1−α)} = \sqrt{8n(1/2 + \epsilon)(1/2 − \epsilon)}$$

$$= \sqrt{2n−\Theta(1)} \approx \Theta(\sqrt{n}).$$

Hence, by replacing $n$ by $m−1$ and using Lemma 6.2, one can verify:

$$Pr\left(\sum_{i=1}^{k} w_i < \sum_{i=k+1}^{m} w_j\right) = \Omega\left(\frac{1}{\sqrt{m}}\right).$$

ACKNOWLEDGEMENT

This work was supported in part by DARPA RADICS under contract #FA-8750-16-C-0054, NSF grants CCF-1423100 and CCF-1703925, funding from the U.S. DOE OE as part of the DOE Grid Modernization Initiative, and DTRA grant HDTRA1-13-1-0021.

REFERENCES

[1] DARPA Rapid Attack Detection, Isolation and Characterization Systems (RADICS). http://goo.gl/5Erifw.

[2] IEEE benchmark systems. Available at http://www.ee.washington.edu/research/ptca/.

[3] Analysis of the cyber attack on the Ukrainian power grid, Mar. 2016. http://www.nerc.com/pa/CEIS/ISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf.

[4] R. Albert, H. Jeong, and A. L. Barabási. Error and attack tolerance of complex networks. Nature, 406(6794):378–382, 2000.

[5] R. Bapat. Graphs and matrices. Springer, 2010.

[6] A. R. Bergen and V. Vittal. Power Systems Analysis. Prentice-Hall, 1999.

[7] A. Bernstein, D. Bienstock, D. Hay, M. Uzunoglu, and G. Zussman. Sensitivity analysis of the power grid vulnerability to large-scale cascading failures. ACM SIGMETRICS Perform. Eval. Rev., 40(3):33–37, 2012.

[8] D. Bienstock. Electrical Transmission System Cascades and Vulnerability: An Operations Research Viewpoint. SIAM, 2016.

[9] J. A. Bondy and U. Murty. Graph theory, volume 244 of graduate texts in mathematics, 2008.

[10] S. Ciavarella, N. Bartolini, H. Khamfroush, and T. La Forte. Progressive damage assessment and network recovery after massive failures. In Proc. IEEE INFOCOM’17, May 2017.

[11] G. Dán and H. Sandberg. Stealth attacks and protection schemes for state estimators in power systems. In Proc. IEEE SmartGridComm’10, 2010.

[12] R. Deng, P. Zhaung, and H. Liang. Ccpa: Coordinated cyber-physical attacks and countermeasures in smart grid. IEEE Trans. Smart Grid, 2017.

[13] I. Dobson, B. Carreras, V. Lynch, and D. Newman. Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization. Chaos, 17(2):026103, 2007.

[14] M. Garcia, T. Catanach, S. Vander Wiel, R. Bent, and E. Lawrence. Line outage localization using phasor measurement data in transient state. IEEE Trans. Power Syst., 31(4):3019–3027, 2016.

[15] M. R. Garey and D. S. Johnson. Computers and intractability: a guide to the theory of NP-completeness. 1979.

[16] K. Khandeparkar, P. Patre, S. Jain, K. Ramamritham, and R. Gupta. Efficient PMU data dissemination in smart grid. In Proc. ACM e-Energy’14 (poster description), June 2014.

[17] J. Kim and L. Tong. On topology attack of a smart grid: undetectable attacks and countermeasures. IEEE J. Sel. Areas Commun, 31(7):1294–1305, 2013.

[18] J. Kim, L. Tong, and B. J. Thomas. Subspace methods for data attack on state estimation: A data driven approach. IEEE Trans. Signal Process., 63(5):1102–1114, 2015.

[19] T. Kim, S. J. Wright, D. Biestock, and S. Harnett. Efficient PMU data dissemination in smart grid. In Proc. ACM e-Energy’14 (poster description), June 2014.

[20] J. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. In Proc. ACM-SIAM SDM’04, June 2004.

[21] S. Li, Y. Yilmaz, and X. Wang. Quickest detection of false data injection attack in wide-area smart grids. IEEE Trans. Smart Grid, 6(6):2725–2735, 2015.

[22] Z. Li, M. Shahidehpour, A. Alabdulwahab, and A. Abusorrah. Bilevel model for analyzing coordinated cyber-physical attacks on power systems. IEEE Trans. Smart Grid, 7(5):2260–2272, 2016.

[23] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali. Distributed optimal load shedding for disaster recovery in smart electric power grids: A second-order approach. In Proc. ACM SIGMETRICS’14 (poster description), June 2014.

[24] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. ACM Trans. Inf. Syst. Secur., 14(1):13, 2011.

[25] F. J. MacWilliams and N. J. A. Sloane. The theory of error-correcting codes. Elsevier, 1977.

[26] N. M. Manousakis, G. N. Korres, and P. S. Georgilakis. Taxonomy of PMU placement methodologies. IEEE Trans. Power Syst., 27(2):1070–1077, 2012.

[27] T. Nesti, J. Nair, and B. Zwart. Reliability of DC power grids under uncertainty: a large deviations approach. arXiv preprint arXiv:1606.02986, 2016.

[28] C. Phillips. The network inhibition problem. In Proc. ACM STOC’93, May 1993.

[29] A. Pinar, J. Meza, V. Donde, and B. Lesieutre. Optimization strategies for the vulnerability analysis of the electric power grid. SIAM J. Optimiz., 20(4):1786–1810, 2010.

[30] S. Soltan, M. Yannakakis, and G. Zussman. Joint cyber and physical attacks on power grids: Graph theoretical approaches for information recovery. In Proc. ACM SIGMETRICS’15, June 2015.

[31] S. Soltan and G. Zussman. Power grid state estimation after a cyber-physical attack under the AC power flow model. In Proc. IEEE PES-GM’17, 2017.

[32] J. E. Tate and T. J. Overbye. Line outage detection using phasor angle measurements. IEEE Trans. Power Syst., 23(4):1644–1652, 2008.

[33] J. E. Tate and T. J. Overbye. Double line outage detection using phasor angle measurements. In Proc. IEEE PES’09, July 2009.

[34] D. Z. Tootaghai, H. Khamfroush, N. Bartolini, S. Ciavarella, S. Hayes, and T. La Forte. Network recovery from massive failures under uncertain knowledge of damages. In Proc. IFIP Networking’17, June 2017.

[35] O. Vukovic, K. C. Sou, G. Dán, and H. Sandberg. Network-layer protection schemes against stealth attacks on state estimators in power systems. In Proc. IEEE SmartGridComm’11, 2011.

[36] J. Zhang and L. Sankar. Physical system consequences of unobservable state-and-topology cyber-physical attacks. IEEE Trans. Smart Grid, 7(4), 2016.

[37] Y. Zhao, J. Chen, A. Goldsmith, and H. V. Poor. Identification of outages in power systems with uncertain states and optimal sensor locations. IEEE J. Sel. Topics Signal Process., 8(6):1140–1153, 2014.

[38] Y. Zhao, A. Goldsmith, and H. V. Poor. On PMU location selection for line outage detection in wide-area transmission networks. In Proc. IEEE PES’12, July 2012.

[39] H. Zhu and G. B. Giannakis. Sparse overcomplete representations for efficient identification of power line outages. IEEE Trans. Power Syst., 27(4):2215–2224, 2012.

[40] H. Zhu and Y. Shi. Phasor measurement unit placement for identifying power line outages in wide-area transmission system monitoring. In HICSS’14, pages 2483–2492, 2014.