Low Latency Detection of Sparse False Data Injections in Smart Grids

Israel Akingeneye, Student Member, IEEE and Jingxian Wu, Senior Member, IEEE

Abstract—We study low latency detections of sparse false data injection attacks in power grids, where an adversary can maliciously manipulate power grid operations by modifying measurements at a small number of smart meters. When a power grid is under attack, the detection delay, which is defined as the time difference between the occurrence and detection of the attack, is critical to power grid operations. A shorter detection delay can shorten the response time, thus prevent catastrophic impacts from the attack. The objective of this paper is to develop low latency false data detection algorithms that can minimize the detection delay subject to constraints on false alarm probability. The false data injection can be modeled with a sparse attack vector, with each non-zero element corresponding to one meter under attack. Since neither the support nor the values of the sparse attack vector is known, a new orthogonal matching pursuit cumulative sum (OMP-CUSUM) algorithm is proposed to identify the meters under attack while minimizing the detection delay.

In order to recover the support of the sparse vector, we develop a new stopping condition for the iterative OMP algorithm by analyzing the statistical properties of the power grid measurements. Theoretical analysis and simulation results show that the proposed OMP-CUSUM algorithm can efficiently identify the meters under attack, and reliably detect false data injections with low delays while maintaining good detection accuracy.

Index Terms—low latency detection, orthogonal matching pursuit, false data injection, cumulative sum.

I. INTRODUCTION

Smart grid is a combination of power infrastructure, smart meters, and a network of computers [1]. Compared to traditional power grids, smart grid is more robust and efficient owing to the improvement in energy management, control, and system monitoring enabled by the incorporation of networks of computers and smart meters. This, though, comes with a price of grid security and privacy.

Attackers can exploit the cyber-infrastructure of the grid to launch cyber-attacks that can compromise normal grid operations. Some malicious party can launch a cyber-attack by modifying the measurement results obtained by the supervisory control and data acquisition (SCADA) system, such as the power injected or flowing on different buses, and the phase angle of the voltage phasors at different buses. False data injected in the measurement results will affect the real time control of grid operations, thus cause significant damages to power grids. In [2], it is demonstrated that an attacker can take advantage of the configuration of the power system by compromising a small number of meters. The cyber-attacks can be performed by breaking into the communication network of the SCADA system, or by remotely accessing the automation devices such as the remote terminal units (RTU) installed at the substations [3].

A large number of methods have been developed to detect various forms of cyber-attacks in smart grids [2], [4], [5], [6], [7]. Most of these methods rely on residual based detections, where the detection is performed by analyzing the difference between the estimated and actual power measurements. In addition, almost all existing detection methods are developed to improve detection accuracy or state observability, with little or no attention given to detection delay, which is defined as the time difference between the occurrence and detection of cyber-attacks. Detection delay of cyber-attacks is crucial to the stability and operations of power grids. A longer detection delay might comprise the entire power grids and cause power loss to millions of people. On the other hand, a lower detection delay can shorten the response time, such that remedial actions and/or counter measures can be taken to significantly reduce the damages and economic losses caused by cyber-attacks.

Low latency detection can be performed by employing theories from quickest change detection (QCD), which is designed to detect a change in the statistical distribution of a random process [8], [9]. The time instant of the occurrence of the change in distribution is denoted as a change point. The objective of QCD is to minimize the detection delay of the change point under the constraints of an upper bound on probability of false alarm (PFA) or a lower bound on average run length (ARL). QCD can be classified into two categories: Bayesian and non-Bayesian change detections. For Bayesian change detection such as the well known Shiryaev procedure [8], the change point is modeled as a random variable, and Bayesian detection methods rely on knowledge of the prior distribution of the change point. When the change point prior distribution is unknown, we can resort to non-Bayesian methods such as the cumulative sum (CUSUM) test [10], which follows the min-max criterion to minimize the detection delay under the worst case change point distribution.

Both Bayesian and non-Bayesian QCD methods require precise knowledge of the statistical distributions of the random process before and after the change. However, it might be difficult, if not impossible, to obtain the exact distributions in many
practical applications, especially the post-change distribution that usually corresponds to abnormal operation conditions. In case of false data injection, it is impossible to obtain the exact post-change distribution, which depends on the unknown attack vector. In [11], the classical CUSUM algorithm is extended with the generalized likelihood ratio test (GLRT), which estimates the unknown parameter in the distribution through maximum likelihood estimation.

There are limited works on low latency detection of false data injection in smart grids. A generalized CUSUM detector is proposed in [12] for false data detection, where the GLRT is utilized to estimate the unknown parameters. The complexity of the generalized CUSUM detector grows exponentially with the number of meters. The complexity mainly arises from the need to identify the meters under attack. A low complexity approximation of the generalized CUSUM is developed in [12], where each meter tracks the false data injection separately. In [13], [14], an adaptive multi-thread CUSUM algorithm is proposed for false data detection in power grids. It is pointed out in [14] that the complexity of GLRT might be too high for practical implementation, thus the Rao test is used for unknown parameter estimation. The elements in the attack vector are assumed to be positive in [13], and such assumption is not always true in practical attacks.

For a large power grid with a large number of buses and meters, it is extremely difficult, if not impossible, for an attacker to attack all meters at once. In almost all cases the attacker can modify the measurements from a small number of meters, that is, the attack is sparse among meters [4]. In recognition of the sparse nature of false data injections, we propose a new orthogonal matching pursuit (OMP) CUSUM algorithm, which utilizes sparse recovery to identify the meters under attack. In the OMP-CUSUM algorithm, the attack vector is modeled as a sparse vector with dimension equal to the number of power measurements in the grid. The indices of the non-zero elements of the attack vector correspond to meters under attack, and the number of non-zero elements is called the sparsity level. A naive way to locate the meters under attack will be to perform exhaustive search of all possible combinations of attack patterns with GLRT, the complexity of which grows exponentially with the number of buses. To reduce complexity, we resort to the OMP algorithm [15], [16], [17], [18], which is a well known algorithm for sparse signal recovery. Given the fact that neither the sparsity nor the support of the attack vector is known, we develop a new stopping condition for the OMP algorithm by analyzing the statistical properties of the measurements in the grid. The stopping condition can accurately terminate the iterative OMP procedure once all meters under attack are successfully identified, without the prior knowledge of the sparsity level. The results of the OMP are then used in the CUSUM algorithm to minimize the detection delay of false data injection, subject to constraints on the detection accuracy and probability of false alarm. The OMP algorithm and CUSUM is combined in an iterative and sequential manner, that is, for each new group of measurements, OMP is used to estimate the support of the attack vector, and the results are then used for the sequential CUSUM test. Theoretical analysis and simulation results show that the newly proposed OMP-CUSUM algorithm can efficiently and promptly detect false data injections with low complexity, low detection delays, and good detection accuracy.

The remainder of this paper is organized as follows. The system model and problem formulation are described in Section II. In Section III, we study the low latency attack detection problem using CUSUM test, and highlight the high computational complexity of GLRT-based CUSUM. The OMP-CUSUM algorithm is presented in Section IV. In Section V, we develop a worst case attack vector, which will be used to test the performance of the proposed algorithm. Simulation results are given in Section VI, and Section VII concludes this paper.

II. PROBLEM FORMULATION

A. System Model

We consider a power system with \( n + 1 \) buses. Each bus is equipped with a meter measuring the power flow and power injections. Without loss of generality, we will only consider a system model of active power flows and power injections. Define the set of buses connected to bus \( i \) as \( \mathcal{X}_i \) with cardinality \( c_i = |\mathcal{X}_i| \). Denote the power injection into bus \( i \) as \( P_i \), and the power flow from bus \( i \) to bus \( j \) as \( P_{ij} \), \( \forall \ j \in \mathcal{X}_i \). The SCADA system provides a total of \( m = m_1 + m_2 \) measurements, where \( m_1 = n + 1 \) is the number of power injections and \( m_2 = \frac{1}{2} \sum_{i=1}^{n+1} |\mathcal{X}_i| \) is the number of power flows. Define the power measurement vector as \( z = [z_1, z_2, \ldots, z_m]^T \in \mathbb{R}^{m \times 1} \), where \( (\cdot)^T \) is the matrix transpose operator and \( \mathbb{R} \) is the set of real numbers.

In phase measurement, one of the \( n + 1 \) buses will serve as a reference, and we only need to measure or estimate the phases of the remaining \( n \) buses relative to that of the reference bus. Without loss of generality, assume that the \( (n+1) \)-th bus is the reference, and define the phase vector of the remaining \( n \) buses as \( x = [x_1, x_2, \ldots, x_n]^T \), where \( x_i \) is the phase of the \( i \)-th bus.

The relationship between the observation vector \( z_l \) and the state vector \( x \) can be expressed as

\[
    z = h(x) + e, \quad (1)
\]

where \( e = [e_1, e_2, \ldots, e_m]^T \in \mathbb{R}^{m \times 1} \) is the measurement error vector at the sampling instant \( l \), and \( h(x) = [h_1(x), \ldots, h_m(x)]^T \) is a function of phase angles.

In this paper we use the standard DC power flow model [19], which results in a linear approximation of the model in (1) as

\[
    z = Hx + e, \quad (2)
\]

where \( H \in \mathbb{R}^{m \times n} \) is the measurement Jacobian matrix for the real power flow and power injection measurements. As in [4], we assume that both the state variables \( x \) and measurement noise \( e_l \) are zero-mean Gaussian with covariance matrices \( \Sigma_x \) and \( \Sigma_e \), respectively. That is, \( e \sim \mathcal{N}(0, \sigma^2_e I_m) \) and \( x \sim \mathcal{N}(0, \sigma^2_x I_n) \), where \( I_m \) is a size-\( m \) identity matrix and
\[ \sigma^2_e \text{ and } \sigma^2_x \text{ are the variances of } e \text{ and } x, \text{ respectively. It is not hard to see that } z_i \text{ is Gaussian distributed with zero mean and covariance matrix } \Sigma_z = \sigma^2_e H^T H + \sigma^2_x I_m. \]

Based on the observations in (2), the state estimator can obtain an estimate \( \hat{x} \) of the state variable \( x \), such that the mean squared error (MSE) \( \sigma^2_0 = E[|\hat{x} - x|^2] \) between the estimated and the actual state variables is minimized. This can be achieved with the minimum mean squared error (MMSE) estimator [4], \( \hat{x} = Kz \), where \( K = \Sigma_z H^T \Sigma_x^{-1} \).

The adversary’s intention is to mislead the state estimator into making more estimation errors by modifying the power measurements at certain meters. This could lead to wrong decisions by the control center, which may decide to increase/decrease power injections at certain buses in the system based on the faulty state estimate.

**B. Problem Formulation**

An intruder can launch an attack on certain meter readings and intentionally modify the measurements corresponding to these meters. Assume attack happens at time \( \theta \) and it modifies the measurements on \( s < m \) meters. The attack vector can thus be modeled by using a \( s \)-sparse attack vector \( a \) of dimension \( m \), which has \( s \) non-zero values corresponding to the \( s \) meters under attack. The observed measurement vector at the sampling instant \( t \) is

\[
z_t = \begin{cases} 
  Hx + e, & \text{if } t < \theta \\
  Hx + a + e, & \text{if } t > \theta 
\end{cases}
\]

(3)

Under the Bayesian setting, the attack time \( \theta \) is modeled as a random variable with prior probability \( Pr(\theta = k) = \pi_k \), for \( k = 1, 2, \ldots \). We want to detect the attack as soon as it occurs, subject to certain performance constraints, such as the probability of false alarm. The detection is performed by using all historical measurement data \( z_1, z_2, \ldots, z_l \) up to this moment \( l \). To this end, we define the detection procedure \( \delta \) as a mapping from the observed measurement sequence \( z^{1:l} = \{z_1, z_2, \ldots, z_l\} \) to a positive integer as

\[
\delta : z^{1:l} \rightarrow \{k : k \leq l\}, l = 1, 2, \ldots
\]

(4)

The estimated attack time is thus \( \delta(z^{1:l}) = \hat{\theta} \leq l \) for some \( l \).

Following the detection procedure in (4), we define, respectively, the probability of false alarm (PFA) and the average detection delay (ADD) as

\[
PFA(\delta) = Pr(\hat{\theta} < \theta),
\]

and

\[
ADD(\delta) = E[\hat{\theta} - \theta | \hat{\theta} < \theta],
\]

(6)

The problem of quickest detection aims to minimize the average detection delay under the constraint of an upper bound of the probability of false alarm or a lower bound on the average run length to false alarm.

Solving the problem requires knowledge of the distributions of the observed measurements before and after the attack. From (3), define the null hypothesis \( H_0 \), which corresponds to the distribution before the attack, and the alternative hypothesis \( H_1 \), which corresponds to the distribution after the attack, as

\[
H_0 : z_i \sim N(0, \Sigma_x), \quad H_1 : z_i \sim N(a, \Sigma_x), \quad \|a\|_0 = s.
\]

(7)

where \( \|a\|_0 \) is the \( \ell_0 \) norm that returns the number of non-zero elements in \( a \).

Denote the distributions of \( z_i \) before and after the attack as \( f_0(z_i) \) and \( f_1(z_i|a) \), respectively. It should be noted that the attack vector \( a \) is unknown at the receiver. Thus the post-attack distribution is unknown.

**III. Quickest Detection With Unknown Attack Vector**

In this section, we develop the quickest detection algorithm with an unknown attack vector. The quickest detection algorithm is developed by extending the CUSUM procedure [20] with GLRT.

First we will formulate the CUSUM procedure by assuming that the attack vector \( a \) is known. Then we will extend the CUSUM procedure to the case with unknown attack vector.

If the attack vector \( a \) is known, then the likelihood ratio (LR) at time instant \( l \) can be calculated as

\[
\lambda_l = \frac{f_1(z_l | a)}{f_0(z_l)} = \exp \left( z_l^T \Sigma_x^{-1} a - \frac{1}{2} a^T \Sigma_x^{-1} a \right).
\]

(8)

Define the cumulative log-likelihood ratio (LLR) of the samples \( z^{k:l} = \{z_k, z_{k+1}, \ldots, z_l\} \) as

\[
\eta_{k:l} = \log \left( \prod_{i=k}^l \lambda_i \right) = \sum_{i=k}^l (z_i^T \Sigma_x^{-1} a - \frac{1}{2} a^T \Sigma_x^{-1} a).
\]

(9)

With the cumulative LLR given in (9), the CUSUM procedure with known attack vector can be written as [10]

\[
\delta = \inf \left\{ l : C_l \geq B \right\}, \quad \text{with } C_l = \max_{1 \leq k \leq l} \eta_{k:l}
\]

(10)

where \( B \) is the threshold chosen such that the constraint on the false alarm probability is satisfied.

The test statistics \( C_l \) can be recursively calculated as

\[
C_l = \max (0, C_{l-1}) + \log \lambda_l
\]

with \( C_0 = 0 \).

The classical CUSUM procedure in (10) requires the knowledge of the attack vector \( a \). In practice, \( a \) is unknown at the detector. As a result, we cannot directly calculate the cumulative LLR \( \eta_{k:l} \) or the test statistics \( C_l \). This problem can be solved by using GLRT, where we estimate the value of \( a \) by maximizing the cumulative log-likelihood ratio (GLR) as [4]

\[
\hat{a}_{k,l} = \arg\max_{a \in \Omega_s} \sum_{i=k}^l (z_i^T \Sigma_x^{-1} a - \frac{1}{2} a^T \Sigma_x^{-1} a). \quad \text{(12)}
\]

where \( \Omega_s \) is the set of all \( s \)-sparse attack vectors.

For a length-\( m \) \( s \)-sparse vector \( a \), there are \( Q_s = \binom{m}{s} \) sparse patterns. For the \( q \)-th sparse pattern, denote the indices
of the non-zero elements as \(k_{q,1} < k_{q,2} < \cdots < k_{q,s_q}\), for \(q = 1, \ldots, Q_s\). If the attack vector assumes the \(q\)-th sparse pattern, then removing the zero elements in \(a_q = [a_{q,1}, a_{q,2}, \ldots, a_{q,s_q}]^T\). The cumulative GLR in (9) can be alternatively written as

\[
\eta_{k:l}^{(q)} = \sum_{i=k}^{l} (z_i^T \Lambda_q a_q - \frac{1}{2} a_q^T \Phi_q a_q).
\]  

(13)

where \(\Lambda_q\) is a \(m \times s\) submatrix of \(\Sigma_z^{-1}\), and it is obtained by removing the \(m-s\) columns with indices corresponding to the zero elements in the \(q\)-th sparse pattern. Similarly, \(\Phi_q\) is a \(s \times s\) submatrix of \(\Sigma_z^{-1}\), and it is obtained by removing the \(m-s\) rows and columns with indices corresponding to the zero elements in the \(q\)-th sparse pattern.

Define

\[
\eta_{k:l}^{(q)} = \sup_{a_q \in \mathbb{R}^s} \sum_{i=k}^{l} (z_i^T \Lambda_q a_q - \frac{1}{2} a_q^T \Phi_q a_q)
\]

(14)

The CUSUM with GLR can thus be represented as

\[
\delta = \inf \left\{ l : \max_{1 \leq k \leq l} \max_{q=1,2,\ldots,Q_s} \eta_{k:l}^{(q)} \geq B \right\}.
\]

(15)

The above quickest detection algorithm requires the exhaustive search of all \(m_Q = 2^m - 1\) sparse patterns, and the exhaustive search needs to be performed for each value of \(1 \leq k \leq l\). The complexity grows exponentially with \(m\) and it becomes prohibitively high when \(m\) is large. A low complexity OMP-CUSUM algorithm is proposed in the next section to balance the tradeoff between complexity and performance.

IV. ORTHOGONAL MATCHING PURSUIT-CUSUM (OMP-CUSUM) TEST

A low complexity OMP-CUSUM algorithm is proposed in this section to balance the tradeoff between complexity and performance. Instead of performing exhaustive search over all sparse patterns, we propose to adopt the OMP algorithm [15] [16] and modify it for the CUSUM test. The OMP algorithm will be used to identify the sparse attack vector that can maximize the cumulative GLR as in (12).

In order to employ the OMP in the cumulative GLR calculation, we need to rewrite the optimization problem in (12) in the form of a linear optimization. The result is given as follows.

**Lemma 1.** The optimization problem in (12) can be alternatively expressed as

\[
\begin{align*}
\min_{A} & \quad \frac{1}{2} \|y - Aa\|_2^2 \\
\text{s.t.} & \quad \|a\|_0 = s
\end{align*}
\]

(16)

(17)

where \(\|b\|_2 = \sqrt{b^T b}\) is the \(\ell_2\)-norm of a vector, \(\|a\|_0\) is the \(\ell_0\)-norm, \(A = D^{-\frac{1}{2}} U\), \(D\) is a diagonal matrix with the eigenvalues of \(\Sigma_z\) on its main diagonal, \(U\) is the corresponding orthonormal eigenvector matrix, that is, \(\Sigma_z = U^T DU\), and

\[
y = \frac{1}{l-k+1} A \sum_{i=k}^{l} z_i.
\]

(18)

**Proof.** The proof is in Appendix A. \(\square\)

We propose to solve the problem in Lemma 1 by using OMP. The basic idea of OMP is to sequentially identify the columns of \(A\) that has the strongest correlation with the vector \(y\), given the fact that \(y\) is a linear combination of the columns of \(A\) corresponding to the non-zero elements of the sparse vector \(a\).

We first describe the OMP algorithm when the sparsity level \(s\) is known. The results are then used to develop an OMP algorithm with unknown sparsity level.

**A. OMP with Known Sparsity Level**

Based on the optimization problem in Lemma 1, the OMP algorithm is described as follows.

- **Step 1.** Initialize the residual \(r_0 = y\) and the iteration counter \(t = 1\). Initialize the set of non-zero index vector as \(I_0 = \emptyset\). Define the index set \(I_0 = \{1,2,\ldots,m\}\).

- **Step 2.** At the \(t\)-th iteration, find the column of \(A\) that has the maximum absolute inner product with the residual \(r_{t-1}\) as

\[
\hat{r}_t = \arg \max_{a \in I_0 \cup I_{t-1}} |r_{t-1}^T A_j|,
\]

(19)

where \(A_j\) denotes the \(j\)-th column of \(A\). Update \(I_t = \{\hat{r}_t\} \cup I_{t-1}\). Denote \(A_{I_t}\) as a submatrix of \(A\) consisting the columns \(A_i\) with \(i \in I_t\).

- **Step 3.** Update the residual \(r_t\) by projecting \(y\) onto the null space of \(A_{I_t}\)

\[
r_t = (I_m - P_{I_t}) y
\]

(20)

where \(I_m\) is a size \(m\) identity matrix, \(P_{I_t} = A_{I_t}(A_{I_t}^T A_{I_t})^{-1} A_{I_t}^T\) is the projection onto the linear space spanned by the columns of \(A_{I_t}\).

- **Step 4.** Set \(t = t + 1\), and go back to step 2 until the stopping conditions are met. If the sparsity level \(s\) is known, we can stop at the \(s\)-th iteration. When the sparsity level is unknown, the stopping condition will be discussed in the next subsection.

- **Step 5.** If the stopping conditions are met, then stop and output an estimate of \(a\) by solving the following optimization problem

\[
\hat{a} = \arg \max_{\|a\|_0 = s} \|y - A_{I_t} a\|_2^2 = (A_{I_t}^T A_{I_t})^{-1} A_{I_t}^T y
\]

(21)

At time instant \(l\), we need to perform the OMP algorithm for each \(1 \leq k \leq l\) to identify the attack vector \(\hat{a}_{k,l}\). Once \(\hat{a}_{k,l}\) is identified, then we can update the cumulative GLR as

\[
\hat{\eta}_{k:l} = \sum_{i=k}^{l} \left( z_i^T \Sigma_z^{-1} \hat{a}_{k,l} - \frac{1}{2} (\hat{a}_{k,l})^T \Sigma_z^{-1} \hat{a}_{k,l} \right)
\]

(22)

With the cumulative GLR defined in (22), the CUSUM with GLR can then be written as

\[
\delta = \inf \left\{ l : \max_{1 \leq k \leq l} \hat{\eta}_{k:l} \geq B \right\}.
\]

(23)
In some sparse sensing applications such as those in [15], [16] where the signal sparsity is known, the above algorithm stops in Step 4 when the iteration $t = s$. In contrast, for our problem, the detector has no knowledge of the sparsity level, $s$, of the attack vector, which is crucial to the performance of the OMP algorithm. In the next sub-section, we solve this problem by developing new stopping conditions for the OMP algorithm based on the residual analysis.

### B. OMP with Unknown Sparsity Level

We propose to develop the stopping condition of the OMP algorithm by analyzing the statistical properties of the residual $r_t$ in (20) at each iteration $t$. From (3), (18), and (20), the residual can be written as

$$r_t = \frac{1}{L} P_t^\bot A \sum_{i=k}^l z_i$$  \hspace{1cm} (24)$$

where $P_t^\bot = I_m - P_t \in \mathbb{R}^{m \times m}$ projects to the null space of the column space spanned by $A_{Z_t}$, $L = l - k + 1$, and $z_i$ is the observation vector defined in (3).

Denote the true support set of $a$ as $I^*_t$, that is, the elements of $I^*_t$ are the indices of the non-zero elements of $a$. If the support set of the attack vector is successfully recovered at the $t$-th iteration, that is, $I^*_t \subseteq I_t$, then the residual $r_t$ does not contain any information of $a$. We denote this as the null hypothesis $H_0$, and the OMP algorithm should stop once $H_0$ is detected. On the other hand, if $I^*_t \cap I_t \neq I^*_t$, that is, the index of at least one non-zero element of $a$ is not in $I_t$, then $r_t$ still depends on $a$. This is denoted as the alternative hypothesis $H_1$, and the algorithm needs to continue to the next iteration under $H_1$.

From (3) and (24), the hypothesis test on the residual $r_t$ can be written as

$$H_0 : r_t = \frac{1}{L} P_t^\bot A \sum_{i=k}^l v_i$$

$$H_1 : r_t = \frac{1}{L} P_t^\bot A \sum_{i=k}^l (v_i + a), \text{ if } a \neq 0,$$  \hspace{1cm} (25)$$

where $v_i = Hx_i + e_i \sim \mathcal{N}(0, \Sigma_x)$.

Since $P_t^\bot \in \mathbb{R}^{m \times m}$ projects to the null space of $A_{Z_t} \in \mathbb{R}^{m \times t}$, which has a column rank of $t$, the rank of $P_t^\bot$ is $m - t$. Due to row-redundancy deficiency of the matrix $P_t^\bot$, the residual $r_t$ in (25) is a degenerate Gaussian distribution. However, as illustrated in [21], we can formulate a full rank sub-matrix $C_t \in \mathbb{R}^{(m-t) \times m}$ by choosing $m - t$ arbitrary rows of $P_t^\bot$. Without loss of generality, we formulate $C_t \in \mathbb{R}^{(m-t) \times m}$ by using the first $m - t$ rows of $P_t^\bot$. Then the hypothesis in (25) can be reformulated as

$$H_0 : \tilde{r}_t = \frac{1}{L} C_t A \sum_{i=k}^l v_i \sim \mathcal{N}(0, \Sigma_t)$$

$$H_1 : \tilde{r}_t = \frac{1}{L} C_t A \sum_{i=k}^l (v_i + a) \sim \mathcal{N}(\mu_t, \Sigma_t),$$  \hspace{1cm} (26)$$

We have the following lemma regarding the distribution of $\tilde{r}_t$ under the null and alternative hypotheses, respectively.

**Lemma 2.** The reduced residual vector $\tilde{r}_t$ is Gaussian distributed under both the null and alternative hypotheses, and

$$\tilde{r}_t|H_0 \sim \mathcal{N}(0, \Sigma_t)$$

$$\tilde{r}_t|H_1 \sim \mathcal{N}(\mu_t, \Sigma_t)$$  \hspace{1cm} (27)$$

where $\Sigma_t = \frac{1}{L} C_t C_t^T$, $\mu_t = \frac{L}{T} C_t A a$, and $L$ is the number of corrupted measurements out of the $L$ collected measurement samples.

**Proof.** The proof is in Appendix B.

It should be noted that the value of $\rho$ used in Lemma 2 is unknown, and we do not need $\rho$ for the stopping condition developed in this subsection.

We can perform likelihood ratio test (LRT) at Step 4 of the OMP algorithm to detect between the null and alternative hypothesis. If $H_0$ is detected at iteration $t$, then the algorithm stops. Otherwise we move on to the next iteration. The development of the optimum decision rule requires the statistical distribution of $y$, which is given in the following lemma.

**Lemma 3.** The distributions of $y = \frac{1}{L} A \sum_{i=k}^l z_i$ under the null and alternative hypotheses are, respectively,

$$y|H_0 \sim \mathcal{N}\left(0, \frac{1}{L} I_m\right)$$

$$y|H_1 \sim \mathcal{N}\left(Aa, \frac{1}{L} I_m\right).$$  \hspace{1cm} (28)$$

**Proof.** The proof is in Appendix C.

**Theorem 1.** For a given probability of false alarm $\sigma$, the OMP algorithm with unknown sparsity stops at the $t$-th iteration if the following condition is met

$$T_t = y^T C_t^T \Sigma_t^{-1} C_t y < \lambda_t.$$  \hspace{1cm} (29)$$

The threshold $\lambda_t$ is calculated as a function of the probability of false positive $\sigma = \Pr(T > \lambda_t|H_0)$ as

$$\lambda_t = 2\Gamma^{-1}\left(\frac{m - t}{2}, \sigma \Gamma\left(\frac{m - t}{2}\right)\right),$$  \hspace{1cm} (30)$$

where $\Gamma(m) = \int_0^\infty x^{m-1} \exp(-x)dx$ is the Gamma function, $\Gamma(M, b) = \int_b^\infty y^{M-1} \exp(-y)dy$ is the upper incomplete Gamma function, and $\Gamma^{-1}(M, y)$ is its inverse.

**Proof.** The proof is in Appendix D.

### V. OPTIMUM ATTACK VECTOR FROM ADVERSARY’S PERSPECTIVE

In order to evaluate the performance of the proposed algorithm, we design a worst case attack vector that is difficult to detect, but can cause large damage to the system. Then we can evaluate the performance of the proposed OMP-CUSUM algorithm by using the worst case attack vector.

For the CUSUM procedure, the average detection delay is asymptotically inversely proportional to the Kullback-Leibler distribution of $\tilde{r}_t$ under both the null and alternative hypotheses.
The probability of stopping can be calculated as [22]

\[ D(f_1 \| f_0) = \frac{1}{2} \left[ \text{trace} \left( \Sigma_x \Sigma_x^{-1} \right) + a^T \Sigma_x^{-1} a - m \right] \ln \left( \frac{\| \Sigma_x \Sigma_x^{-1} \|}{\| \Sigma_x \|} \right) \]

\[ = \frac{1}{2} a^T \Sigma_x^{-1} a. \]  

(31)

In general, it is difficult for an attacker to gain access to every meter in the system. Instead, the adversary might have access to a subset of \( s \) meters with indices \( T_s = \{i_1, i_2, \cdots, i_s\} \). Denote \( a_s = a_{T_s} \), which contains the elements \( a_k \) with \( k \in T_s \). Thus the KL divergence can be rewritten as

\[ D(f_1 \| f_0) = \frac{1}{2} a_s^T \Phi_s a_s \]  

(32)

where \( \Phi_s \) is an \( s \times s \) submatrix of \( \Sigma_x^{-1} \), and it contains the rows and columns of \( \Sigma_x^{-1} \) with indices in \( T_s \).

The damage caused by the attack vector can be measured by the energy of the attack, or equivalently, the additional mean square error of state estimation due to the attack. From [4], the energy of the attack can be calculated as

\[ \sigma_a^2 = \| \Sigma_x H^T \Sigma_x^{-1} a \|^2 = \| \Sigma_x H^T \Lambda_s a_s \|^2 = \| K_s a_s \|^2. \]  

(33)

where \( \Lambda_s \) contains the columns of \( \Sigma_x^{-1} \) with indices in \( T_s \), and \( K_s = \Sigma_x H^T \Lambda_s \).

We can then design the worst case attack vector by solving the following optimization problem.

\[ \min_{a_s \in \mathbb{I}_s} \quad a_s^T \Phi_s a_s \]  

s.t. \( \| K_s a_s \|^2 \geq \gamma \),

\[ \text{(34)} \]

where \( \gamma \) is the minimum attack energy desired by the adversary. A similar approach, but with different objective function, has been taken in [4], where the attack vector is designed to minimize the estimation residue error subject to the constraint on a lower bound of the attack energy.

The optimization problem in (34) can be solved analytically, and the solution is given as follows.

**Corollary 1.** The optimum attack vector that solves the optimization problem in (34) is

\[ a^*_s = \sqrt{\frac{\gamma}{\| K_s u_{\min} \|^2}} u_{\min} \]  

(35)

where \( u_{\min} \) is the generalized eigenvector corresponding to the minimum eigenvalue of the matrix pair \( (\Phi_s, K_s^T K_s) \).

**Proof.** The proof is shown in Appendix E.

---

**VI. SIMULATION RESULTS**

In this section, we present the simulation results by using several standard IEEE bus configurations. The simulations are performed by using the MATLAB power system software [23]. In the simulations, it is assumed that the state variables \( x \) are Gaussian distributed with zero mean and covariance matrix \( \Sigma_x = \sigma_x^2 I_m \). The covariance matrices of the measurement noise is \( \Sigma_c = \sigma_c^2 I_m \). The signal-to-noise ratio (SNR) in dB is defined as \( 10 \log \frac{\sigma_x^2}{\sigma_c^2} \). The change point is assumed to follow a geometric distribution with parameter \( p_0 \), that is, \( p_k = (1 - p_0)k^{-1}p_0 \). In all simulations, we set \( \text{SNR} = 10 \text{ dB} \) and \( p_0 = 0.1 \).

Fig. 1 shows the probability that the OMP algorithm meets the stopping conditions in Theorem 1 during various iterations as a function of the probability of false positive (PFP) \( \alpha \) for the IEEE 14-bus system. The PFP represents the probability that the OMP algorithm does not stop after the full recovery of the attack vector support. The attack vector is randomly generated with sparsity \( s = 5 \) and then scaled as in (35) with \( \gamma = \sigma_x^2 = 0.0217 \). Each point on the curves in Fig. 1 was obtained by running 10,000 trials. As predicted by our theoretical analysis, the probability that the OMP algorithm stops increases with the number of iterations. With \( s = 5 \), ideally the algorithm should stop at the 5-th iteration. Stopping at \( t < 5 \) iterations means that some attacks are not identified. In this experiment, the probability of stopping is relatively low (less than 0.27 at PFP = 0.05) during the first four iterations because the residual vector (20) still contains non-zero components of the attack vector. On the other hand, the stopping probability rises significantly after the 4th iterations. At PFP = 0.05, the probabilities of stopping at the 5th, 6th, and 7th iterations are 0.89, 0.93, and 0.95, respectively.

The average detection delays are shown as functions of the false alarm probability for the IEEE 14-bus and 57-bus systems in Figs. 2 and 3, respectively. The attack vectors are designed in the same way as those used in Fig. 1 with \( \gamma = 0.0217 \). As in [13], the threshold \( B \) in (10) is obtained by fixing the probability of false alarm, PFA = \( \beta \), and setting \( B = \log \frac{\beta^{-1}}{p_c} \). The probability of false positive for the OMP...
stopping conditions is $\sigma = 0.01$. Each point in the figures was obtained through Monte Carlo simulations with 4,000 trials. The average detection delay is represented as the number of observation samples. For both the 14- and 57-bus systems, the ADD decreases as the sparsity $s$ becomes smaller, mainly due to the fact that a smaller $s$ results in a better recovery rate of the support of the attack vector. As expected, the ADD is a monotonic decreasing function in PFA for all system configurations. At PFA = 0.05, the ADDs of the 14-bus system with $s = 2, 5$, and $6$ are 0.9, 2.4, and 4.1 samples, respectively; the ADDs of the 57-bus system with $s = 2, 6$, and 15 are 1.04, 1.32, and 1.95 samples, respectively. Therefore the algorithms can detect various attacks with low latency and high accuracy.

Fig. 4 shows the average detection delay as a function of the normalized attack energy $\gamma$. Two different attack vectors are considered. One is the optimum attack vector as designed in Corollary 1, and the other one is a random attack vector normalized to meet the attack energy constraint. For the random attack vector, each point in the curve is obtained by averaging over 10,000 different realization of the random attack vectors. The probability of false alarm is set as $\beta = 0.072$, and the sparsity level is $s = 2$. All other parameters are the same as in Fig. 2. The proposed OMP-CUSUM algorithm can reliably and quickly detect both types of attack vectors, but the optimum attack vector is more difficult to detect than the random one. For instance, at the average detection delay of 1 sample, the minimum detectable normalized attack energy of the optimum and random attack vectors are -14.5 dB and -16.7 dB, respectively.

VII. CONCLUSION

Low latency detections of false data injected to smart grids have been studied in this paper. Motivated by the fact that a malicious party usually can only attack a small number of meters, we have developed a new OMP-CUSUM algorithm for low latency detection of false data injections. Unlike conventional CUSUM algorithm that relies on the knowledge of the attack vector, the OMP-CUSUM algorithm can efficiently identify the meters under attack, and minimize the detection delay of bad data injection under the constraint of the probability of false alarm. A new stopping condition has been proposed for the OMP algorithm to identify the meters under attack. An optimum attack vector has also been developed to test and validate the performance of the proposed algorithm. Simulation results have shown that the proposed OMP-CUSUM algorithm can accurately and reliably detect intrusions with small delays and low complexities, and the detection performance improves as the sparsity level of the attack vector decreases.

APPENDIX A

PROOF OF LEMMA 1

Define $w_{k:l} = \frac{1}{l-k+1} \sum_{i=k}^{l} z_i$. Then the objective function in (12) can be alternatively written as

$$J = (l - k + 1) \left( w_{k:l}^T \Sigma_z^{-1} a - \frac{1}{2} a^T \Sigma_z^{-1} a \right)$$  (36)
Based on the eigenvalue decomposition of $\Sigma_z$, we have
\[
\Sigma_z^{-1} = U^T D^{-1} U = A^T A
\]  
(37)
In addition, $y = Aw_{k:t}$.

Then (36) can be alternatively represented as
\[
\frac{2J}{l - k + 1} = 2y^T Aa - a^T A^T y y + y^T y
\]
(38)
Finding $a$ to maximize $J$ is thus equivalent to minimize $\|y - Aa\|^2$. This completes the proof.

**APPENDIX B**
**PROOF OF LEMMA 2**

Since $v_i \sim N(0, \Sigma_z)$ and $\tilde{r}_t$ is a linear combination of $v_i$, $\tilde{r}_t$ is Gaussian distributed under both the null and alternative hypotheses.

Under $H_1$, the mean of $\tilde{r}_t$ is $E[\tilde{r}_t|H_1] = \frac{1}{L} C_t A \Sigma_z \sum_{i=k}^l a = \frac{\hat{C}_t A a}{L} \Sigma_z \tilde{r}_t$.

The covariance matrices under both $H_0$ and $H_1$ are
\[
\Sigma_t = E[\tilde{r}_t \tilde{r}_T^T | H_0] = \frac{1}{L} C_t A \Sigma_z A^T C_t = \frac{1}{L} C_t C_t^T
\]
(40)
where the last equality is based on the fact that $A = D^{-\frac{1}{2}} U$ and $\Sigma_z = U^T D U$. This completes the proof.

**APPENDIX C**
**PROOF OF LEMMA 3**

Since $y$ is the linear combination of $z_i$, which is Gaussian distributed, $y$ is Gaussian distributed under both the null and alternative hypotheses.

Under the null hypothesis, we have $z_i = v_i \sim N(0, \Sigma_z)$. Given that $z_i$ and $z_j$ are independent for $i \neq j$, the covariance matrix of $y$ is
\[
E[y y^T | H_0] = \frac{1}{L^2} A \sum_{i=k}^l \sum_{j=k}^l E[z_i z_j^T] A^T
\]
(41)
\[
= \frac{1}{L^2} \sum_{i=k}^l A \Sigma_z A^T = \frac{1}{L} I_m
\]
(42)
Under the alternative hypothesis, we have $z_i = v_i + a \sim N(a, \Sigma_z)$. Thus $E[y y^T | H_1] = \frac{1}{L} A \Sigma_z \sum_{i=k}^l a = A a$. The covariance matrix under the alternative hypothesis can be derived in a similar manner as (41). This completes our proof.

**APPENDIX D**
**PROOF OF THEOREM 1**

The LLR for the hypothesis test in (26) is
\[
\log L(\tilde{r}_t, a) = \log \frac{Pr(\tilde{r}_t|a, H_1)}{Pr(\tilde{r}_t|H_0)}
\]
\[
= \tilde{r}_t^T \Sigma_t^{-1} \mu_t - \frac{1}{2} \mu_t^T \Sigma_t^{-1} \mu_t
\]
(43)
where $\mu_t = \frac{a}{L} C_t A a$. Since $a$ is unknown, we can perform GLRT, where $a$ can be estimated by maximizing the LLR. Setting $\frac{\partial \log L(\tilde{r}_t, a)}{\partial a} = 0$ yields
\[
\frac{\rho}{L} C_t A \hat{a} = \tilde{r}_t,
\]
(44)
where $\hat{a}$ is the maximum likelihood (ML) estimate of $a$. Replacing $a$ in (43) with $\hat{a}$ in (44) results in
\[
\log L(\tilde{r}_t, \hat{a}) = \frac{1}{2} \tilde{r}_t^T \Sigma_t^{-1} \tilde{r}_t
\]
(45)
Given that $\tilde{r}_t = \frac{1}{L} C_t A \sum_{i=k}^l z_i = C_t y$ and $\Sigma_t = \frac{1}{L} C_t C_t^T$, the LLR can be written as
\[
\log L(\tilde{r}_t, \hat{a}) = \frac{L}{2} y^T C_t^T (C_t C_t^T)^{-1} C_t y
\]
(46)
It is apparent that $C_t^T (C_t C_t^T)^{-1} C_t$ is a projection matrix that projects to the linear space spanned by the $(m - t)$ rows of $C_t$, thus it can be represented as
\[
C_t^T (C_t C_t^T)^{-1} C_t = V \cdot \text{Diag} [1_{m-t}; 0] \cdot V^T
\]
(47)
where $\text{Diag} [1_{m-t}; 0]$ is a $m \times m$ diagonal matrix with the vector $[1_{m-t}; 0]$ on its main diagonal, that is, the first $m-t$ elements of the main diagonal are 1s, and all the rest are 0s.

Define $w = \sqrt{TV} y$. Since $y$ is Gaussian distributed as in Lemma 3, it can be easily shown that
\[
w | H_0 \sim N(0, I_m)
\]
\[
w | H_1 \sim N \left( \sqrt{TV} A a, I_m \right)
\]
(48)
The LLR in (46) can then be alternatively written as
\[
\log L(\tilde{r}_t, \hat{a}) = \frac{1}{2} w \cdot \text{Diag} [1_{m-t}; 0] \cdot w
\]
\[
= \frac{1}{2} \sum_{k=1}^{m-t} w_k^2
\]
(49)
From (49), the hypothesis test can be represented as
\[
T_t = y^T C_t^T \Sigma_t^{-1} C_t y = L y C_t^T (C_t C_t^T)^{-1} C_t y
\]
\[
= \sum_{k=1}^{m-t} w_k^2 \tilde{\lambda}_t
\]
(50)
where the threshold $\lambda_t$ will be determined based on the probability of false positive $\sigma$.

Given the distribution of $w$ in (48), it can be easily shown that $T_t$ follows the $\chi^2$ distribution with $m-t$ degrees of freedom under $H_0$ and the non-central $\chi^2$ distribution with $m-t$ degrees of freedom under $H_1$.

Therefore, the distributions of $T_t$ under $H_0$ is
\[
f_0(x | H_0) = \frac{x^{m-t-1} \exp(-\frac{x}{2})}{2^{\frac{m-t}{2}} \Gamma (\frac{m-t}{2})}
\]
The probability of false positive is
\[
\sigma = \Pr (T > \lambda_t | H_0) = \int_{\lambda_t}^{\infty} f_0(x | H_0) dx
\]
\[
= \frac{1}{\Gamma (\frac{m-t}{2})} \Gamma \left( \frac{m-t-\lambda_t}{2} \right)
\]
(51)
The above equation can then be used to obtain the threshold in (30). It should be noted that the threshold does not require the knowledge of $a$. This completes the proof.
Appendix E

Proof of Corollary 1

The problem is a quadratically constrained quadratic program (QCQP), thus it has zero duality gap and can be solved by using its dual problem. The Lagrangian of the optimization problem in (34) is

\[
L(a_s, \lambda) = a_s^T \Phi_s a_s - \lambda \left( \|K_s a_s\|^2 - \gamma \right) \tag{52}
\]

\[
= a_s^T \Phi_s a_s - \lambda K_s^T K_s a_s + \lambda \gamma \tag{53}
\]

If \(\Phi_s - \lambda K_s^T K_s\) is positive semidefinite, i.e., \(\Phi_s - \lambda K_s^T K_s \succeq 0\), then \(L(a_s, \lambda)\) is convex. It can be minimized by setting \(\frac{\partial L(a_s, \lambda)}{a_s} = 0\), and the solution is

\[
\Phi_s a_s = \lambda K_s^T K_s a_s \tag{54}
\]

From (54), \(\lambda\) must be a generalized eigenvalue of \((\Phi_s, K_s^T K_s)\).

From (52) and (54), the minimum Lagrangian is \(\lambda \gamma\) when \(\Phi_s - \lambda I_s \succeq 0\).

Thus the dual function of the optimization problem in (34) is

\[
g(\lambda) = \inf_{a_s} L(a_s, \lambda) = \begin{cases} \lambda \gamma, & \text{if } \Phi_s - \lambda K_s^T K_s \succeq 0 \\ -\infty, & \text{if } \Phi_s - \lambda K_s^T K_s \prec 0 \end{cases} \tag{55}
\]

The dual problem of (34) can then be written as

\[
\max_{a_s \in \Omega} \lambda \gamma \quad \text{s.t. } \Phi_s - \lambda K_s^T K_s \succeq 0, \tag{56}
\]

The maximum \(\lambda\) that satisfies \(\Phi_s - \lambda K_s^T K_s \succeq 0\) is \(\lambda_{\min}\), which is the minimum generalized eigenvalue of \((\Phi_s, K_s^T K_s)\).

Substituting \(\lambda\) with \(\lambda_{\min}\) in (54), we can see that the optimal attack vector \(a^*_s\) should be in the form \(a^*_s = c \cdot u_{\min}\), where \(u_{\min}\) is the generalized eigenvector corresponding to \(\lambda_{\min}\), and \(c\) is a constant used to ensure that \(\|K_s a^*_s\|^2 = \gamma\). Solving \(\|K_s c u_{\min}\|^2 = \gamma\) yields (35).

References


