Knowl Inf Syst (2017) 52:815–834
DOI 10.1007/s10115-016-1019-8

REGULAR PAPER

False data separation for data security in smart grids

Hao Huang[1] · Qian Yan[1] · Yao Zhao[2] · Wei Lu[3] · Zhenguang Liu[4] · Zongpeng Li[2,5]

Received: 17 May 2016 / Accepted: 28 December 2016 / Published online: 18 January 2017

Abstract The smart grid is emerging as an efficient paradigm for electric power generation, transmission, and consumption, based on optimized decision making and control that leverage the measurement data of sensors and meters in the grid. False data injection is a new type of power grid attacks aiming to tamper such important data. For the security and robustness of the grid, it is critical to separate the false data injected by such attacks and recover the original measurement data. Nonetheless, the existing approaches often neglect the true changes on original measurement data that are caused by the real perturbations on grid states and hence have a risk of removing these true changes as injected false data during the data recovery. In this paper, we preserve these true changes by modeling the false data problem as a rank-bounded L_1 norm optimization and propose both offline and online algorithms to filter out the

✉ Wei Lu
lu-wei@ruc.edu.cn

Hao Huang
haohuang@whu.edu.cn

Qian Yan
qy@whu.edu.cn

Yao Zhao
yaozhao@ucalgary.ca

Zhenguang Liu
lzg@nus.edu.sg

Zongpeng Li
zongpeng@ucalgary.ca

1 State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China

2 Department of Computer Science, University of Calgary, Calgary, Canada

3 School of Information and DEKE, MOE, Renmin University of China, Beijing, China

4 School of Computing, National University of Singapore, Singapore, Singapore

5 School of Computer, Wuhan University, Wuhan, China

injected false data and recover original measurement data. Trace-driven simulations verify the efficacy of our solution.

Keywords False data separation · Data recovery · Data security · Optimization · Grassmann manifold

1 Introduction

In a modern smart grid, as illustrated in Fig. 1, electricity is produced in generating plants and transmitted to power stations via transmission lines. Following voltage reduction at distribution stations, electricity is further carried to customers through a distribution network. To ensure the security of such a power grid, decision-making units as exemplified by the Energy Control Center (ECC) require accurate, up-to-date information on the grid states, including voltage profiles, current of power flows, grid frequency evolvement, and load profiles. To this end, ECC collects these state information through the Supervisory Control and Data Acquisition (SCADA) system [1], which monitors the grid by taking a set of measurements every several seconds or minutes for current grid state. State estimator in the control center estimates the grid states through analysis of collected measurement data and power system models, where bad data caused by measurement noise/error and malicious attacks are filtered out.

The wide adoption of digital control and communication technologies provides smart grid operators unprecedentedly abundant information on the status of devices in a power grid. However, such information collection and transmission, including smart meter deployment, are often based on Internet technologies and broadband communications that are prone to security attacks. The US power grid has reportedly fallen victim to cyber intrusions in the past [17]. Even if state estimators work appropriately and provide an accurate snapshot of the grid's health status, the measurement data collected by ECC are still prone to attacks by

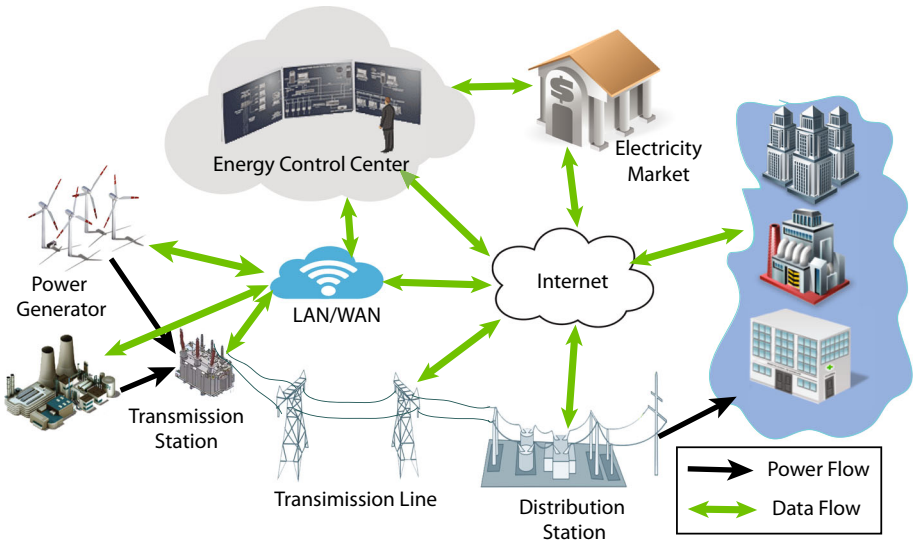

Fig. 1 Cyber-physical infrastructure of a smart grid

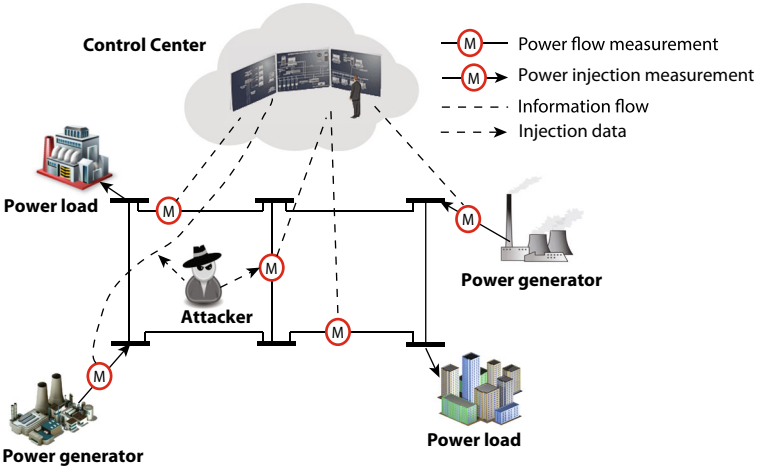

Fig. 2 An illustration of false data injection attack in a 5-bus system: An attacker may distort meter reading to mislead the control center

malicious users, often with strong economic or political motivations. Attackers may hack meters in the grid and manipulate them to conduct false data injection attacks, polluting the data collected by ECC. Figure 2 illustrates an example of false data injection attacks.

This work aims to separate the injected false data from original real measurement data. In the grid, the measurement data collected by ECC from meters in each time slot form a *measurement vector*. The measurement vectors over multiple time slots form a *measurement matrix*. The real measurement matrix is often subject to intrinsic temporal correlation of grid states [13] and measurement topology [6]. The spatial correlation of power sources and loads further increases the correlation among real measurement vectors [14]. Such high correlation across measurement vectors leads to *a low-rank structure* of the real measurement matrix. On the other hand, the injected false data matrix over time is *sparse*, since an attacker often has limited access to resources that are necessary to compromise a large number of measurement and transmission units. Therefore, the measurement matrix observed at ECC is usually the sum of a low-rank real measurement matrix and a sparse malicious data matrix. Given this conclusion, most of the existing work formulated the problem of false data separation as an optimization problem of minimizing the rank of the real measurement matrix and the L_0 norm (i.e., the number of nonzero entries) of the injected false data matrix. Nonetheless, in this way, the rank of real measurement matrix would be underestimated. This is because there are often perturbations on grid states at different time slots and thus a few changes between the real measurement vectors. These changes will slightly bring down the correlations among real measurement vectors, resulting in a slightly higher rank for the real measurement matrix.

To avoid the flaw of the existing work, in our problem formulation, we still minimize the L_0 norm of the injected false data attack matrix, but only confine the rank of the real measurement matrix with an upper bound instead of minimizing it. As L_0 norm minimization is NP hard, we resort to a relaxation in L_1 norm with a constraint on the upper bound of the rank of real measurement matrix. Furthermore, we adopt manifold modeling and optimization techniques to translate the rank-constrained matrix optimization into geometric optimization. The key enabling tool is a Grassmann manifold, a smooth geometric object in which each 'point' can be viewed as a subspace spanned by a matrix. Explicitly keeping a matrix low rank is equivalent to implicitly moving a point on a Grassmann manifold. Furthermore, we

apply a gradient descent method to track the subspace spanned by the real measurement matrix on the Grassmann manifold.

We present both offline and online versions of our approach to false data separation. The offline algorithm requires collecting all the measurement vectors over time instances to recover the real measurement matrix. The online version conducts subspace updating continuously as measurements arrive, and progressively refines its output. To the best of our knowledge, it is the first time that we explicitly solve the problem of online false data separation in a smart grid. For a smart grid that has been proven observable by observability analysis, its energy management system (EMS) can estimate the grid's operation state by a power system measurement model, and our proposed approaches can work as bad data processing methods and be integrated in the bad data processing routine of each state estimation task. Once the EMS carries out a state estimation task, the bad data detection routine will be launched, and our proposed approaches will be executed to identify and remove injected false data in observed measurements. Empirical studies show that our proposed approaches can accurately recover the original measurement matrix from corrupted data.

In the rest of the paper, Sect. 2 reviews related work. Section 3 introduces system model and preliminaries. The offline and online algorithms are presented in Sects. 4 and 5. Section 6 contains simulation studies, and Sect. 7 concludes the paper.

2 Related work

Existing studies [12] show that carefully constructed false data injection attacks can circumvent traditional detection mechanisms and introduce arbitrary errors to power system estimates. A vast literature has hence been devoted to more sophisticated algorithm design that defends false data injection attacks, which can be categorized into three types, namely (1) *protection-based defense*, (2) *false data detection*, and (3) *false data separation*.

Protection-based defense aims to prevent false data injection attacks from being launched in the first place by protecting critical sensors and meters which are carefully selected [2,3,20]. Nevertheless, this line of mechanisms becomes less practical when the number of system states is very large. Protecting a given number of sensors is expensive, and whether perfect protection can always be guaranteed is questionable. Furthermore, such mechanisms require an accurate grid topology, which is hard to obtain and is not always static.

False data detection aims to detect a false data injection attack after it happens. It formulates the problem of state estimation under malicious attacks as a hypothesis testing problem [10,20]. Such mechanism requires a *prior* probability distribution on the grid states. In contrast, our solution is a *prior-free* one and can not only detect but also eliminate false data injected to the smart grid.

False data separation further aims to recover real measurements on the grid states from the corrupted ones by exploiting the low-rank feature of the real measurement matrix and the sparseness feature of the injected false data matrix. As pioneered by the work of Liu et al. [13], the existing approaches to false data separation minimize the rank of real measurement matrix and maximize the sparseness of the injected false data matrix. Nonetheless, these approaches have a risk of underestimation on the rank of real measurement matrix, which may degrade the performance of false data separation.

Besides, a few optimization methods have been proposed for the problem of low-rank and sparse matrix separation, i.e., finding a low-rank matrix $\mathbf{L} \in \mathbb{R}^{m \times n}$ and a sparse matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ such that the sum of $\mathbf{L}$ and $\mathbf{S}$ is equal to a given matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$. This matrix

separation problem is very similar to the problem of false data separation in a smart grid. Hence, these optimization methods can also be applied in the smart grid to identify and separate the false data injected in observed measurements, although they are not originally proposed in the smart grid. According to their adopted objective function for optimization, these optimization methods can be categorized into two main groups, namely (1) *convex optimization-based approaches* and (2) *matrix factorization-based approaches*.

Convex optimization-based approaches address the matrix separation problem by solving the convex optimization problem as below.

$$\min_{\mathbf{L},\mathbf{S}} ||\mathbf{L}||_* + \lambda ||\mathbf{S}||_1, \quad \text{s.t. } \mathbf{D} = \mathbf{L} + \mathbf{S}$$

where $||\mathbf{L}||_*$ refers to the nuclear norm of $\mathbf{L}$, $||\mathbf{S}||_1$ stands for the L_1 norm of $\mathbf{S}$, and $\lambda > 0$ is a weighting factor. The above problem is usually called as robust principal component analysis (rPCA) [5] (or principal component pursuit [7]). Two types of solutions have been developed for this problem. The first one adopts and extends the well-known iterative shrinkage (or soft-thresholding) scheme for convex optimization [5], and second one is known as the augmented Lagrangian alternating direction method (ALADM) [21] which iteratively minimizes the augmented Lagrangian function of rPCA with respect to either $\mathbf{L}$ or $\mathbf{S}$.

Matrix factorization-based approaches express the matrix $\mathbf{L} \in \mathbb{R}^{m \times n}$ as a matrix product $\mathbf{L} = \mathbf{V}\Lambda$, where $\mathbf{V} \in \mathbb{R}^{m \times r}$ and $\Lambda \in \mathbb{R}^{r \times n}$ $[r < \max(m, n)]$, and translate the problem of low-rank and sparse matrix separation as the form below.

$$\min_{\mathbf{V},\Lambda,\mathbf{S}} ||\mathbf{S}||_1, \quad \text{s.t. } \mathbf{D} = \mathbf{V}\Lambda + \mathbf{S}$$

Similar ideas have been applied to matrix compressed sensing problem [8] and matrix completion problem [18]. To minimize the variables $\mathbf{V}$ and Λ which only appear in the objective function, several types of approaches have been proposed, such as the classic GS (Gauss–Seidel) scheme [8], the efficient SOR (Successive Over-Relaxation) scheme [18], and more recently an ALADM-like scheme [16] with least square method for variable update.

3 System model and preliminaries

3.1 State estimation in smart grids

The state estimation problem is to estimate the state of a smart grid from redundant measurement data collected from meters. In the grid, the linearized model of state measurement can be expressed as a linear regression equation as follows.

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ refers to the *true state vector* of the grid, and n is the number state variables; $\mathbf{z} = (z_1, z_2, \ldots, z_m)^{\mathrm{T}}$ denotes the *measurement vector* observed at ECC, and m is the number of meters deployed; $\mathbf{H} = (h_{i,j})_{m \times n}$ is a constant *Jacobian matrix* that links meter readings to real grid states and is determined by grid topology and line impedances; and $\mathbf{e}$ represents the *measurement error*, which is usually modeled as a zero-mean Gaussian noise vector with covariance matrix $\mathbf{R}$ [19].

Based on the equation above, traditional approaches estimate the true state vector as follows.

$$\hat{\mathbf{x}} = (\mathbf{H}^{\mathrm{T}}\mathbf{R}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}\mathbf{z}$$

With the estimated vector $\hat{\mathbf{x}}$ for the true grid state, the state estimator in ECC can compute a measurement residual $(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}})$ to check whether there are large measurement errors caused by system faults or malicious attacks [12]. The answer will be yes if the L_2-norm of this residual is greater than a threshold τ, i.e., if $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||_2 > \tau$.

3.2 False data injection attacks

A false data injection attack could successfully introduce errors into true state variables by exploiting the configuration of the smart grid while bypassing a norm threshold-based detection method [12]. To this end, it may inject a false data vector $\mathbf{a} = (a_1, a_2, \ldots, a_m)^{\mathrm{T}}$, which can be expressed as a linear combination of column vectors of $\mathbf{H}$, i.e., $\mathbf{a} = \mathbf{Hc}$ where $\mathbf{c}$ contains combination coefficients. Then, the measurement vector observed at ECC is as follows.

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a} = \mathbf{H}(\mathbf{x} + \mathbf{c}) + \mathbf{e}$$

Let $\hat{\mathbf{x}_a}$ and $\hat{\mathbf{x}}$ be the estimates of $\mathbf{x}$ with the distorted and true data, respectively. Then, $\hat{\mathbf{x}_a} = \hat{\mathbf{x}} + \mathbf{c}$, and we have

$$||\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_a|| = ||\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})|| = ||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}|| < \tau$$

Hence, by setting $\mathbf{a} = \mathbf{Hc}$, attackers can trick the state estimator into believing that the true state vector is $\mathbf{x_a} = \mathbf{x} + \mathbf{c}$.

3.3 Features of true and false data

Let $\mathbf{z_k}$ and $\mathbf{a_k}$ be the true measurement vector and injected false state vector at time t_k, respectively. Matrices $\mathbf{Z_o} = (\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_t})$ and $\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_t})$ represent the true measurements and injected false data over a period of t time slots. Then, the observed measurement matrix $\mathbf{Z_a}$ at ECC is

$$\mathbf{Z_a} = \mathbf{Z_o} + \mathbf{A}.$$

Matrix $\mathbf{Z_o}$ has a *low rank* due to the temporal correlation among measurements over time and the spatial correlation caused by the same measurement topology [6,14]. On the other hand, matrix $\mathbf{A}$ is *sparse* since (i) attackers can compromise only a limited number of meters at a time and (ii) the attack often lasts for a limited time period to avoid detection. Besides the limited resources of attackers, as pointed out by Liu et al. [13], the utilization of Phasor Measurement Units (PMUs), which provide accurate measurements of bus voltage angles and power flows to ECC, can also help reduce the number of comprised measurements in the grid and make the matrix $\mathbf{A}$ sparse. Furthermore, due to their accurate measurements, PMUs are able to minimize the effect of measurement noise and error during the identification of injected false data, and improve the performance of false data detection and separation algorithms.

Given the features of true and false data, one can separate $\mathbf{Z_o}$ from $\mathbf{A}$ in $\mathbf{Z_a}$ by solving the following optimization problem denoted by $P0$.

$$(P0): \min_{\mathbf{Z_o}} \operatorname{rank}(\mathbf{Z_o}) + ||\mathbf{A}||_0, \quad \text{s.t. } \mathbf{Z_a} = \mathbf{Z_o} + \mathbf{A}$$

where the L_0 norm $||\mathbf{A}||_0$ is the number of nonzero entries in $\mathbf{A}$. $P0$ belongs to the class of rank minimization problems, which is NP hard. One may apply robust principle component analysis [5] to approximately solve $P0$ by minimizing a weighted combination of the nuclear

norm of $\mathbf{Z_o}$ and the L_1 norm of $\mathbf{A}$, which is a convex relaxation from the original L_0-norm optimization [13]:

$$\min_{\mathbf{Z_o}} ||\mathbf{Z_o}||_* + \lambda||\mathbf{A}||_1, \quad \text{s.t. } \mathbf{Z_a} = \mathbf{Z_o} + \mathbf{A} \tag{1}$$

where nuclear norm $||\mathbf{Z_o}||_* = \sum_k \sigma_k(\mathbf{Z_o})$, and $\sigma_k(\mathbf{Z_o})$ stands for the kth largest singular value of $\mathbf{Z_o}$.

This kind of nuclear norm-based methods [13] can separate $\mathbf{Z_o}$ and $\mathbf{A}$ with a relatively high probability, but have the following two flaws. (1) Since $P0$ and its approximation Eq. (1) explicitly and implicitly minimize the rank of true measurement matrix $\mathbf{Z_o}$, they may underestimate this rank and cause performance degradation for false data separation. In practice, due to the perturbations on grid states at some time slots, there are a few changes between the real measurement vectors, resulting in a slightly higher rank for $\mathbf{Z_o}$. (2) These existing approaches are available only after measurement vectors are collected over a long time period.

4 Offline false data separation

Aiming at an optimization framework that avoids an underestimation of the rank of $\mathbf{Z_o}$ and naturally supports both offline and online false data separation, we explore a novel solution space inspired from differential geometry. In this section, we start from the offline version, in which we firstly formulate the problem of false data separation as a rank-bounded optimization:

$$\min_{\mathbf{Z_o}} ||\mathbf{Z_a} - \mathbf{Z_o}||_1, \quad \text{s.t. } \text{rank}(\mathbf{Z_o}) \leq r \tag{2}$$

where the upper bound r of $\text{rank}(\mathbf{Z_o})$ can be obtained by a rank estimation strategy [18].

Moreover, let $\mathcal{U}_{m,r} = \{\mathbf{U} \in \mathbb{R}^{m\times r} : \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}_r\}$ be the set of $m \times r$ matrices with r orthonormal columns. The space of $\mathcal{U}_{m,r}$ forms a *Grassmann manifold*, a smooth geometric object in which each 'point' $\mathbf{U} \in \mathcal{U}_{m,r}$ represents a r-D subspace of an m-D vector space. By factorizing $\mathbf{Z_o}$ into two matrices $\mathbf{U} \in \mathcal{U}_{m,r}$ and $\mathbf{W} \in \mathbb{R}^{r\times t}$, the rank constraint in problem (2) can be always satisfied since

$$\text{rank}(\mathbf{Z_o}) = \text{rank}(\mathbf{UW}) \leq \min\big(\text{rank}(\mathbf{U}), \text{rank}(\mathbf{W})\big) \leq r$$

In other words, keeping the matrix $\mathbf{Z_o}$ low rank is equivalent to moving a 'point' on a Grassmann manifold.

Furthermore, we introduce a residual matrix $\mathbf{R} = \mathbf{Z_a} - \mathbf{UW}$, which is also the false data matrix when $\mathbf{UW}$ is recovered exactly. Then, the problem of offline false data separation can be formulated as the optimization problem below.

$$(P1): \min_{\mathbf{U},\mathbf{W}} ||\mathbf{R}||_1, \quad \text{s.t. } \mathbf{UW} + \mathbf{R} - \mathbf{Z_a} = 0$$

$P1$'s augmented Lagrangian function is

$$L(\mathbf{U}, \mathbf{W}, \mathbf{R}, \mathbf{Y}, \rho) = ||\mathbf{R}||_1 + \langle \mathbf{Y}, \mathbf{UW} + \mathbf{R} - \mathbf{Z_a}\rangle + \frac{\rho}{2}||\mathbf{UW} + \mathbf{R} - \mathbf{Z_a}||_2^2$$

where $\mathbf{Y} \in \mathbb{R}^{m\times t}$ is the Lagrange multiplier corresponding to the constraint $\mathbf{UW}+\mathbf{R}-\mathbf{Z_a} = 0$, ρ is the penalty parameter, and $\langle\cdot,\cdot\rangle$ denotes the inner product of two matrices.

$L(\mathbf{U}, \mathbf{W}, \mathbf{R}, \mathbf{Y}, \rho)$ is not jointly convex w.r.t. $(\mathbf{U}, \mathbf{W})$, but it is convex w.r.t. either $\mathbf{U}$ or $\mathbf{W}$ with other one fixed. Given this property, we resort to alternating minimization methods [15] to solve problem $P1$, i.e., iteratively updating either of $\mathbf{U}$ and $\mathbf{W}$ with the other one fixed.

4.1 Updating W with fixed U

To find the optimal $\mathbf{W}$ with fixed $\mathbf{U}$, we employ the alternating direction method of multipliers (ADMM), which is a powerful technique for designing efficient distributed algorithms to solve convex optimization problems [4]. Formally, with current $\mathbf{U}$, matrices $\mathbf{W}$, $\mathbf{R}$, and $\mathbf{Y}$ can be optimized as follows.

$$\begin{aligned}\mathbf{W}_{k+1} &= \arg\min_{\mathbf{W}} L\left(\mathbf{U}, \mathbf{W}_k, \mathbf{R}_k, \mathbf{Y}_k, \rho_k\right)\\ \mathbf{R}_{k+1} &= \arg\min_{\mathbf{R}} L\left(\mathbf{U}, \mathbf{W}_{k+1}, \mathbf{R}_k, \mathbf{Y}_k, \rho_k\right)\\ \mathbf{Y}_{k+1} &= \arg\min_{\mathbf{Y}} L\left(\mathbf{U}, \mathbf{W}_{k+1}, \mathbf{R}_{k+1}, \mathbf{Y}_k, \rho_k\right)\end{aligned}$$

Specifically, the exact expression of the matrices are

$$\begin{aligned}\mathbf{W}_{k+1} &= \left(\mathbf{U}^{\mathrm{T}}\mathbf{U}\right)^{-1}\mathbf{U}^{\mathrm{T}}\left(\mathbf{Z_a} - \mathbf{R}_k - \frac{1}{\rho_k}\mathbf{Y}_k\right)\\ \mathbf{R}_{k+1} &= S_{1/\rho_k}\left(\mathbf{Z_a} - \mathbf{U}\mathbf{W}_{k+1} - \frac{1}{\rho_k}\mathbf{Y}_k\right)\\ \mathbf{Y}_{k+1} &= \mathbf{Y}^k + \rho_k\left(\mathbf{U}\mathbf{W}_{k+1} + \mathbf{R}_{k+1} - \mathbf{Z_a}\right)\end{aligned} \tag{3}$$

where $S_{1/\rho_k}(\cdot)$ is an elementwise soft-thresholding operator which is defined as $S_\tau(x) = \mathrm{sgn}(x)\max(|x| - \tau, 0)$.

4.2 Search optimal U on Grassmann manifold

With the optimal $(\mathbf{W}, \mathbf{Y}, \mathbf{R})$ that minimizes current L, we adopt a gradient descent method to search the optimal $\mathbf{U}$ on the Grassmann manifold. As shown in Fig. 3, the search path follows the manifold's geodesics, and the search direction is determined by function L's descent gradient direction.

The gradient of L over a Grassmann manifold at $\mathbf{U}$ is its *tangent vector* which can be calculated as follows.

$$\nabla L = L_{\mathbf{U}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}L_{\mathbf{U}} \tag{4}$$

where $L_{\mathbf{U}}$ is the derivative of L w.r.t. $\mathbf{U}$ and is calculated as

$$L_{\mathbf{U}} = \mathbf{Y}\mathbf{W}^{\mathrm{T}} + \rho(\mathbf{U}\mathbf{W} + \mathbf{R} - \mathbf{Z_a})\mathbf{W}^{\mathrm{T}}$$

To move closer to optimum of L, we update $\mathbf{U}$ along the descent gradient direction $-\nabla L$. After ℓ iterations, $\mathbf{U}$ will be

$$\mathbf{U}_\ell = \mathbf{U}_{\ell-1}\mathbf{V}\cos\left(\Sigma\theta_\ell\right)\mathbf{V}^{\mathrm{T}} + \mathbf{P}\sin\left(\Sigma\theta_\ell\right)\mathbf{V}^{\mathrm{T}} \tag{5}$$

where $\mathbf{P}\Sigma\mathbf{V}$ is the compact SVD of $-\nabla L$, and θ_ℓ is the step size of moving along the geodesic at the ℓth iteration.

Step size selection Variable step sizes help strike a trade-off between convergence guarantee and speed. Large step sizes are first adopted to quickly reach the neighborhood of the optimum; then, small steps are adopted to avoid overshooting. If the optimum is static, the gradient descent approach is guaranteed to converge to a stationary point as long as the step sizes satisfy the relationship below [11].

$$\lim_{\ell\to+\infty}\theta_\ell = 0 \quad \text{and} \quad \sum_{\ell=1}^{\infty}\theta_\ell = \infty$$

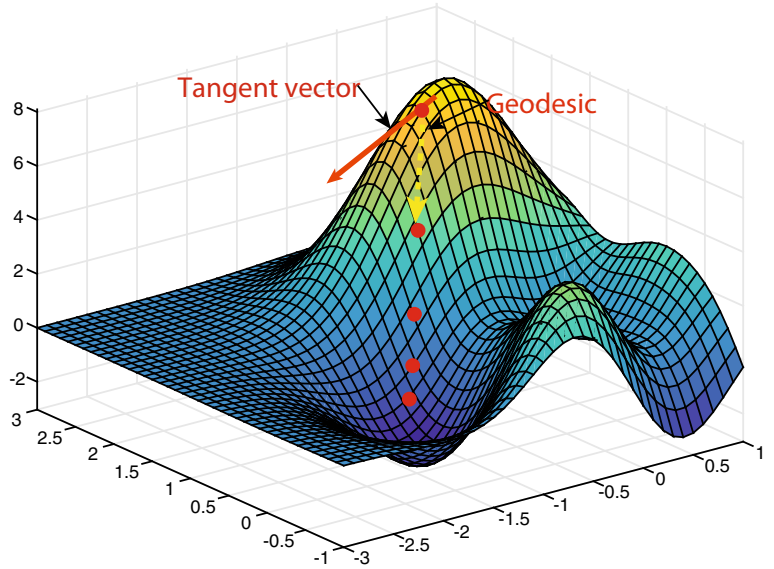

Fig. 3 'Point' evolution on Grassmann manifold. To keep the new 'point' on the manifold, we retract a tangent vector to a geodesic on the manifold for moving the 'point' along

Initial **U** Since Eckart–Young–Mirsky theorem states that the optimum low-rank approximation for a matrix with fit measured by Frobenius norm has an analytic solution in terms of the SVD of the matrix, we use SVD to estimate an initial **U**:

$$\mathbf{Z_a} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i u_i v_i$$

where σ_i is the ith singular value of $\mathbf{Z_a}$, and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. u_i and v_i are the left-singular and right-singular vectors w.r.t. σ_i, respectively. We use the r left-singular vectors (the largest r singular values) as the initial **U**, i.e.,

$$\mathbf{U}_0 = [u_1, u_2, \ldots, u_r].$$

4.3 The offline algorithm

Our offline algorithm (see Algorithm 1) for false data separation takes as inputs the observed measurement matrix $\mathbf{Z_a}$, the initial value $\mathbf{U}_0$ of **U**, the upper bound r for the rank of the true measurement matrix $\mathbf{Z_o}$, and step sizes $\{\theta_\ell\}$. It first estimates the optimal tuple $(\mathbf{W}, \mathbf{R}, \mathbf{Y})$ from the current **U** via the ADMM algorithm (lines 2–7). Given the optimal tuple, a new **U** is estimated by exploiting the gradient descent approach to search a current optimum of **U** on the Grassmann manifold (lines 8–9). The two steps are iteratively performed till convergence. Then, $\mathbf{Z_o} = \mathbf{UW}$ and $\mathbf{A} = \mathbf{Z_a} - \mathbf{Z_o}$ are returned as the true measurement matrix and false data matrix.

4.4 Discussions

Convergence $P1$ is non-convex, hard to be solved efficiently. Algorithm 1 addresses this problem through alternating minimization, which alternates between finding the best **U** and

Algorithm 1 Offline False Data Separation Algorithm

Require: Observed measurement matrix $\mathbf{Z_a}$, $\mathbf{U}_0$, r, $\{\theta_\ell\}$
Ensure: True measurement matrix $\mathbf{Z_o}$, false data matrix $\mathbf{A}$
while UW not converge **do** // alternating minimization
  $\mathbf{Y} = \mathbf{0}$, $\mathbf{R} = \mathbf{0}$, $0 < \alpha < 1.8$, $\rho_0 > 0$, $k = 0$;
  while W not converge **do** // ADMM algorithm
    Update $\mathbf{W}$, $\mathbf{R}$, $\mathbf{Y}$ by Eq. (3);
    $\rho_{k+1} = \alpha\rho_k$;
    $k := k + 1$;
  end while
  Calculate $\bigtriangledown L$ by Eq. (4);
  Iteratively update $\mathbf{U}$ by Eq. (5) till convergence;
end while
$\mathbf{Z_o} = \mathbf{UW}$;
$\mathbf{A} = \mathbf{Z_a} - \mathbf{Z_o}$;

best $\mathbf{W}$. Each alternating step in isolation is convex and tractable. Hence, Algorithm 1 is a typical augmented Lagrangian alternating direction method (ALADM). Nonetheless, to the best of our knowledge, there is no established convergence theory for ALADM algorithms applied to non-convex problems, such as $P1$. On the other hand, extensive empirical evidences, including our experimental results, suggest that ALADM algorithms often have very good convergence behavior [4,13,16]. It is also proved that this kind of approaches will converge to a KKT point under mild conditions. Shen et al. [16] recently showed that the following alternating minimization problem,

$$\min_{\mathbf{U},\mathbf{W},\mathbf{Z}} \quad ||\mathbf{Z} - \mathbf{D}||_1 \quad \text{s.t.} \quad \mathbf{UW} - \mathbf{Z} = \mathbf{0} \tag{6}$$

where $\mathbf{D}$ is a known matrix, has the following property.

Proposition 1 *Let* $\mathbf{X} = (\mathbf{U}, \mathbf{W}, \mathbf{Z})$ *and* $\{\mathbf{X}^j\}_{j=1}^{\infty}$ *be generated through alternating minimization. Assume that* $\{\mathbf{X}^j\}_{j=1}^{\infty}$ *is bounded and* $\lim_{j\to\infty}(\mathbf{X}^{j+1} - \mathbf{X}^j) = 0$. *Then, each accumulation point of* $\{\mathbf{X}^j\}_{j=1}^{\infty}$ *satisfies the KKT conditions of Eq.* (6). *In particular, whenever* $\{\mathbf{X}^j\}_{j=1}^{\infty}$ *converges, it converges to a KKT point of Eq.* (6).

The optimization problem $P1$ has the same structure as Eq. (6). We can hence claim that if variables in $P1$ are always bounded during the alternating minimization, then the convergence result of Algorithm 1 satisfies the necessary conditions (KKT conditions) of being the optimum.

Complexity analysis Executing ADMM takes $O(Kmrt)$ time, where K is the number of iterations in ADMM, and t is the number of vectors collected. The computation of ∇L needs $O(mr^2 + mrt + r^3)$ time, performing SVD of $-\nabla L$ for Eq. (5) requires $O(mrt)$. Since rank $r \ll t$, the overall time complexity of Algorithm 1 is $O\big(N(Kmrt + mrt)\big)$, where N is the number of times for alternating minimization.

Applicability To avoid the flaw of the existing approaches to false data separation, we relax the low-rank assumption on the real measurement matrix $\mathbf{Z_o}$ to be more practical, since real perturbations on grid states at different time slots often lead to a relatively higher rank for $\mathbf{Z_o}$. On the other hand, like the existing work [13,19,20], we still follow the sparseness assumption on the injected false data matrix $\mathbf{A}$, since the attack ability and available resources of an attacker are usually limited for attacking a large smart grid system. In other words, our proposed approaches are designed to defend attack scenarios in which attackers compromise

only a small part of meters in a smart grid. For the cases in which attackers can have access to a significantly large part of meters, we suggest to combine our proposed approach with the protection-based defense methods, such as dynamically encrypting some of the measurements. Then, in order to avoid being detected by the protection-based defense, attackers will try to minimize the number of meters to tamper with, and construct a sparse $\mathbf{A}$, which will be identified and separated by our proposed approach. The above idea is not new, and in fact, it was introduced by Kim et al. [9] as early as in 2011.

5 Online false data separation

An offline false data separation conducts its computation only after collecting a sequence of measurement vectors over a time period. Waiting for such data being accumulated without any action may result in early detection opportunities missed. To defend against false data injection with real-time responses and low computational complexity, we extend our geometry optimization framework to an online version.

Instead of waiting for a complete $\mathbf{Z_a}$ containing observed measurement vectors over a period of t time slots, we evolve the matrix $\mathbf{U} \in \mathcal{U}_{m,r}$ in real time when each observed measurement vector $\mathbf{z_a}_i$ of time i ($i \in \{1, 2, \ldots, t\}$) arrives. Here, $\mathbf{z_a}_i$ is equivalent to the ith column of $\mathbf{Z_a}$. Moreover, let vector $\mathbf{w}_i$ be the ith column of W, and residual vector $\mathbf{r}_i$ be the ith column of the residual matrix $\mathbf{R}$. Then, the rank-bounded optimization in Problem (2) can be reformulated as

$$\min_{\mathbf{U},\mathbf{w}_i} \sum_{i=1}^{t} ||\mathbf{z_a}_i - \mathbf{U}\mathbf{w}_i||_1, \quad \text{s.t. } \mathbf{U} \in \mathcal{U}_{m,r}$$

and our online algorithm solves the problem below.

$$(P2): \min_{\mathbf{U},\mathbf{w}_i} ||\mathbf{r}_i||_1, \quad \text{s.t. } \mathbf{U}\mathbf{w}_i + \mathbf{r}_i - \mathbf{z_a}_i = 0$$

5.1 The online algorithm

We use the same alternating minimization framework to address problem $P2$ since it is non-convex. When $\mathbf{U}$ is fixed as $\mathbf{U}_\ell$, $P2$ is a classic *least absolute deviation* (LAD) problem, which can be naturally solved by ADMM, and the augmented Lagrangian function is as follows.

$$\begin{aligned}\mathcal{L}(\mathbf{w}_i, \mathbf{r}_i, \mathbf{y}_i, \rho_i) = {} & ||\mathbf{r}_i||_1 + \mathbf{y}_i^{\mathrm{T}} (\mathbf{U}_\ell \mathbf{w}_i + \mathbf{r}_i - \mathbf{z_a}_i) \\ & + \frac{\rho_i}{2} ||\mathbf{U}_\ell \mathbf{w}_i + \mathbf{r}_i - \mathbf{z_a}_i||_2^2\end{aligned}$$

where $\mathbf{y}_i$ is the ith Lagrangian multiplier, and ρ_i is the corresponding penalty parameter.

Given $\mathbf{U}_\ell$, the optimal $(\mathbf{w}_i, \mathbf{r}_i, \mathbf{y}_i)$ can be computed to minimize function $\mathcal{L}$ by iteratively refining the triple below.

$$\begin{aligned}\mathbf{w}_i^{k+1} &= \left(\mathbf{U}_\ell^{\mathrm{T}} \mathbf{U}_\ell\right)^{-1} \mathbf{U}_\ell^{\mathrm{T}} \left(\mathbf{z_a}_i - \mathbf{r}_i^k - \frac{1}{\rho_i}\mathbf{y}^k\right) \\ \mathbf{r}_i^{k+1} &= S_{1/\rho_i}\left(\mathbf{z_a}_i - \mathbf{U}_\ell \mathbf{w}_i^{k+1} - \frac{1}{\rho_i}\mathbf{y}^k\right) \\ \mathbf{y}_i^{k+1} &= \mathbf{y}^k + \rho_i \left(\mathbf{U}_\ell \mathbf{w}_i^{k+1} + \mathbf{r}_i^{k+1} - \mathbf{z_a}_i\right)\end{aligned} \tag{7}$$

Algorithm 2 Online False Data Separation Algorithm

Require: Sequence of $\mathbf{z_a}_i$, random unitary matrix $\mathbf{U}$, r, $\{\theta_\ell\}$
Ensure: True measurement matrix $\mathbf{Z_o}$
for $i = 1, 2, \ldots, t$ **do** // $\mathbf{z_a}_i$ of time i arrives
  for $j = 1, 2, \ldots, N$ **do** // alternating minimization
    $\mathbf{y}_i = \mathbf{0}, \mathbf{r}_i = \mathbf{0}, 0 < \alpha < 1.8, \rho_0 > 0, k = 0$;
    while $\mathbf{w}_i$ not converge **do** // ADMM algorithm
      Update $(\mathbf{w}_i, \mathbf{r}_i, \mathbf{y}_i)$ by Eq. (7);
      $\rho_{k+1} = \alpha\rho_k$;
      $k := k + 1$;
    end while
    Iteratively update $\mathbf{U}$ by Eq. (10) till convergence;
  end for
end for
$\mathbf{Z_o} = \mathbf{UW}$; // $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t)$

Then, we apply gradient descent approach to evolve $\mathbf{U}$ from $\mathbf{U}_\ell$ to $\mathbf{U}_{\ell+1}$. To compute function $\mathcal{L}$'s gradient on Grassmann manifold, we first compute the partial derivative of $\mathcal{L}$ w.r.t. $\mathbf{U}$:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{U}} = \mathbf{y}_i\mathbf{w}_i^{\mathrm{T}} + \rho_i\left(\mathbf{U}_\ell\mathbf{w}_i + \mathbf{r}_i - \mathbf{z_a}_i\right)\mathbf{w}_i^{\mathrm{T}} = \left(\mathbf{y}_i - \rho_i\mathbf{e}_i\right)\mathbf{w}_i^{\mathrm{T}}$$

where $\mathbf{e}_i = \mathbf{z_a}_i - \mathbf{U}_\ell\mathbf{w}_i - \mathbf{r}_i$. The gradient $\nabla\mathcal{L}$ of $\mathcal{L}$ is

$$\nabla\mathcal{L} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{U}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{U}} = \left(I - \mathbf{U}\mathbf{U}^{\mathrm{T}}\right)\left(\mathbf{y}_i - \rho_i\mathbf{e}_i\right)\mathbf{w}_i^{\mathrm{T}} = \chi\mathbf{w}_i^{\mathrm{T}}$$

where $\chi = (\mathbf{I} - \mathbf{U}\mathbf{U}^{\mathrm{T}})(\mathbf{y}_i - \rho_i\mathbf{e}_i)$. Along the direction of $-\nabla\mathcal{L}$, $\mathbf{U}$ can be updated to a new value $\mathbf{U}_{\ell+1}$ along the geodesic emanating from $\mathbf{U}_\ell$ on Grassmann manifold:

$$\mathbf{U}_{\ell+1} = \mathbf{U}_\ell\mathbf{V}\cos(\Sigma\theta_\ell)\mathbf{V}^{\mathrm{T}} + \mathbf{P}\sin(\Sigma\theta_\ell)\mathbf{V}^{\mathrm{T}} \tag{8}$$

where $\mathbf{P}\Sigma\mathbf{V}$ is the compact SVD of the descent gradient $-\nabla\mathcal{L}$, and θ_ℓ is the step size of moving $\mathbf{U}_\ell$ along the geodesic.

It is straightforward to compute the gradient using SVD in Eq. (8), since the rank of $\nabla\mathcal{L}$ is 1. The only nonzero singular value is $\sigma = ||\chi|| \cdot ||\mathbf{w}_i||$, and the corresponding singular vectors are $\frac{\chi}{||\chi||}$ and $\frac{\mathbf{w}_i}{||\mathbf{w}_i||}$, respectively. The SVD of $\nabla\mathcal{L}$ can be formulated in the form of matrix product as

$$\nabla\mathcal{L} = \left[\frac{\chi}{||\chi||}, p_2, \ldots, p_r\right]\mathrm{diag}(\sigma, 0, \ldots, 0)\left[\frac{\mathbf{w}_i}{||\mathbf{w}_i||}, q_2, \ldots, q_r\right]^{\mathrm{T}} \tag{9}$$

where $(p_2, \ldots, p_r)$ and $(q_2, \ldots, q_r)$ are arbitrary orthonormal vectors orthogonal to χ and $\mathbf{w}_i$, respectively.

By replacing $\mathbf{V}$ and $\mathbf{P}$ in Eq. (8) with singular vector matrices in Eq. (9), the evolution of $\mathbf{U}$ becomes:

$$\mathbf{U}_{\ell+1} = \mathbf{U}_\ell + \left((\cos(\sigma\theta_\ell) - 1)\mathbf{U}_\ell q_1 - \sin(\sigma\theta_\ell)p_1\right)q_1^{\mathrm{T}} \tag{10}$$

where $p_1 = \frac{\chi}{||\chi||}$ and $q_1 = \frac{\mathbf{w}_i}{||\mathbf{w}_i||}$.

The overall online algorithm is presented in Algorithm 2, in which N is the alternating minimization times to evolve $\mathbf{U}$ for current time slot. Our simulation studies show that the algorithm converges when N is around 20.

5.2 Time complexity

When a new observed measurement vector $\mathbf{z}_{\mathbf{a}i}$ of the ith time slot arrives, estimating the optimal $(\mathbf{w}_i, \mathbf{r}_i, \mathbf{y}_i)$ with ADMM takes $O(Kmr)$ time, where K is the number of iterations in ADMM. Our simulation studies show that K is usually less than 50. Calculating $\nabla\mathcal{L}$ in Eq. (9) requires $O(mr^2)$ time. The update of $\mathbf{U}$ avoids a full SVD computation and is merely a multiplication of a matrix with a vector, which only needs $O(mr)$ time. In summary, the amortized per-slot time complexity of the online algorithm is $O\big(N(Kmr + mr^2)\big)$.

6 Performance evaluation

In this section, we first introduce the experimental setup and then evaluate the accuracy performance of our approaches to false data separation in terms of relative error and ROC criteria. We also discuss the effect of the upper bound r for the real measurement matrix $\mathbf{Z_o}$, followed by an efficiency study for our approaches.

6.1 Experimental setup

We conduct trace-driven empirical studies on two large system models, namely the IEEE 300-bus system and the Polish 3375-bus power system. Each measurement vector consists of the power injection measurements at all buses and power flow data at all branches. These vectors are collected over a period of t time slots at one vector per slot. A small portion ϵ of the state measurements is compromised by false data attackers with arbitrary data. The corrupted data are chosen randomly, and the attacks last a period of Δt.

We evaluate the accuracy performance of our algorithms with the criterion of *relative error* (RE), which is calculated as

$$\text{RE} = \frac{||\mathbf{Z_a} - \mathbf{Z_o}||_F}{||\mathbf{Z_o}||_F},$$

and the technique of *receiver operating characteristic* (ROC) analysis. The hit rate and false alarm rate (FAR) in a ROC graph are defined as

$$\text{Hit Rate} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}}, \quad \text{FAR} = \frac{N_{\text{FP}}}{N_{\text{FP}} + N_{\text{TN}}},$$

where N_{TP}, N_{FN}, N_{FP} and N_{TN} refer to the numbers of true positives, false negatives, false positives, and true negatives, respectively.

We compare our algorithms with the existing two state-of-the-art solutions for false data separation, i.e., the *nuclear norm minimization* (abbreviated as NC) and *low-rank matrix factorization* (abbreviated as LRMF) approaches proposed by Liu et al. [13]. Moreover, since a few optimization methods, such as the rPCA and matrix factorization-based approaches, can be also applied or extended to solve the problem of false data separation, we also compare our algorithms with these methods, including the rPCA approach with iterative shrinkage scheme for optimization (abbreviated as rPCA), and the matrix factorization approach with SOR scheme for optimization (abbreviated as SOR). Note that the NC algorithm is in fact a typical rPCA approach with ALADM scheme for optimization, and the LRMF algorithm is in fact a matrix factorization approach using ALADM scheme. All algorithms in the experiments are implemented in MATLAB, running on a Macbook Pro laptop with 2.4GHz Intel Core i5 CPU and 8GB RAM.

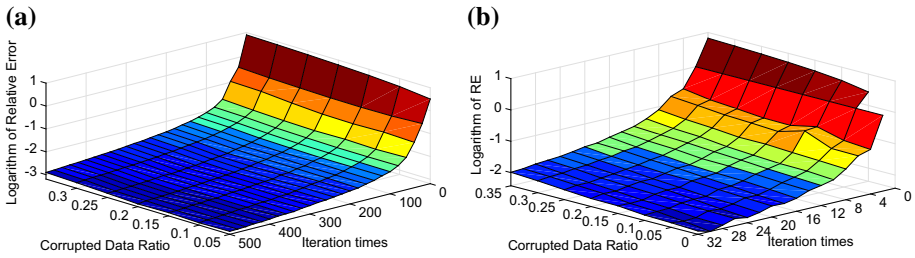

Fig. 4 Accuracy performance of our offline and online algorithms on IEEE 300-bus system. **a** Accuracy performance of offline algorithm, **b** accuracy performance of online algorithm

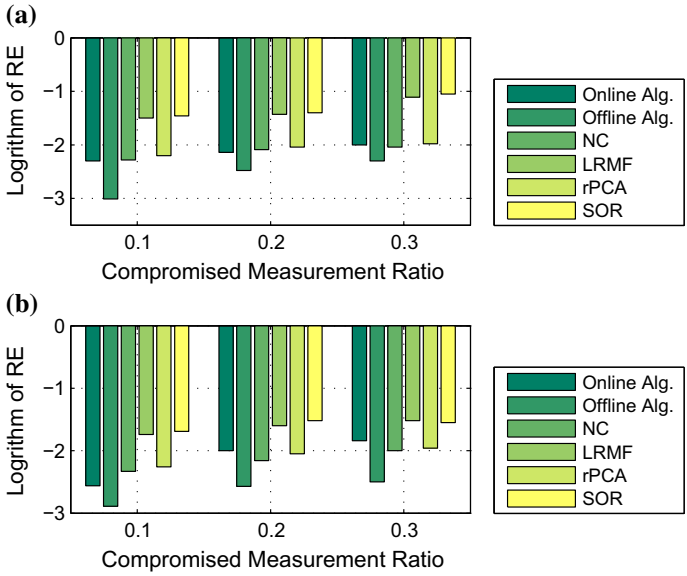

Fig. 5 RE versus compromised data ratio on IEEE 300-bus system and Polish power system. **a** IEEE 300-bus system, **b** Polish power system

6.2 Accuracy performance study

We first study the accuracy performance of our algorithms for recovering true smart grid measurements. We compute the relative error of the recovered results under different corrupted data ratios. Our algorithms are tested on the IEEE 300-bus system, and the measurement data are collected for 300 time instances, which construct an observed measurement matrix $\mathbf{Z_a}$ of size 561×300, as there are 561 measurement variables in the IEEE 300-bus system. In this experiment, we fix the upper bound r for the rank of real measurement matrix at 5 and vary the corrupted data ratio ϵ from 0 to 0.35. Figure 4a shows that after sufficient iterations, our offline algorithm can successfully filter false data with a relative error at level of 10^{-3}, even when 35% of the observed measurements are corrupted; Fig. 4b shows that our online algorithm can converge in less than 32 iterations with precision close to 10^{-3}, when the corrupted data ratio is low. Moreover, from the figure, we can also observe that both of our offline and online algorithms are not sensitive to corrupted data ratio.

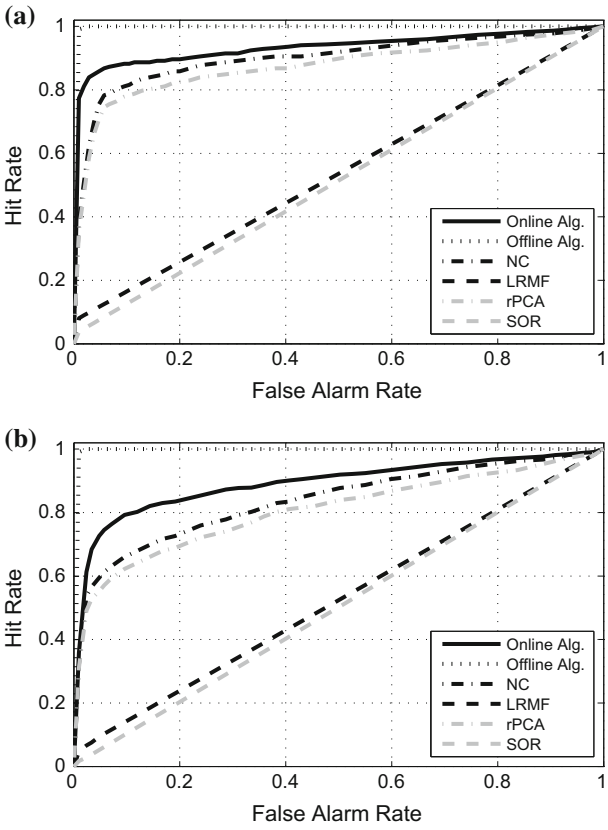

Fig. 6 ROC curves on IEEE 300-bus system and Polish power system. **a** IEEE 300-bus system, **b** Polish power system

The relative error comparison result between our algorithms and the existing approaches is illustrated in Fig. 5. The comparison result on IEEE 300-bus system (with $r = 5$) is shown in Fig. 5a, from which we can obverse that when compromised measurement ratio increases, the performance of our offline algorithm deteriorates slightly, but is still the best; compared with the other five tested algorithms, our online algorithm is relatively less sensitive to the compromised data ratio ϵ. The comparison result on the Polish 3375-bus power system (with $r = 10$) is shown in Fig. 5b and exhibits a similar trend.

Figure 6a, b illustrate the ROC curves (under compromised data ratio $\epsilon = 0.1$) of the six tested algorithms on the IEEE 300-bus system and the Polish 3375-bus power system, respectively. An algorithm has a better performance if it has a high hit rate even when the false alarm rate is low. Figure 6 shows that our offline algorithm has the best performance, and our online algorithm is also better than the NC, LRMF, rPCA, and SOR algorithms.

6.3 Effect of upper bound r for $\mathbf{Z_o}$'s rank

In this experiment, we fix the real rank of real measurement matrix $\mathbf{Z_o}$ on the IEEE 300-bus system to 5, and use different value from 1 to 10 as the upper bound r for the rank of $\mathbf{Z_o}$ recovered by our algorithms. Figure 7a illustrates the effect of different upper bound r on

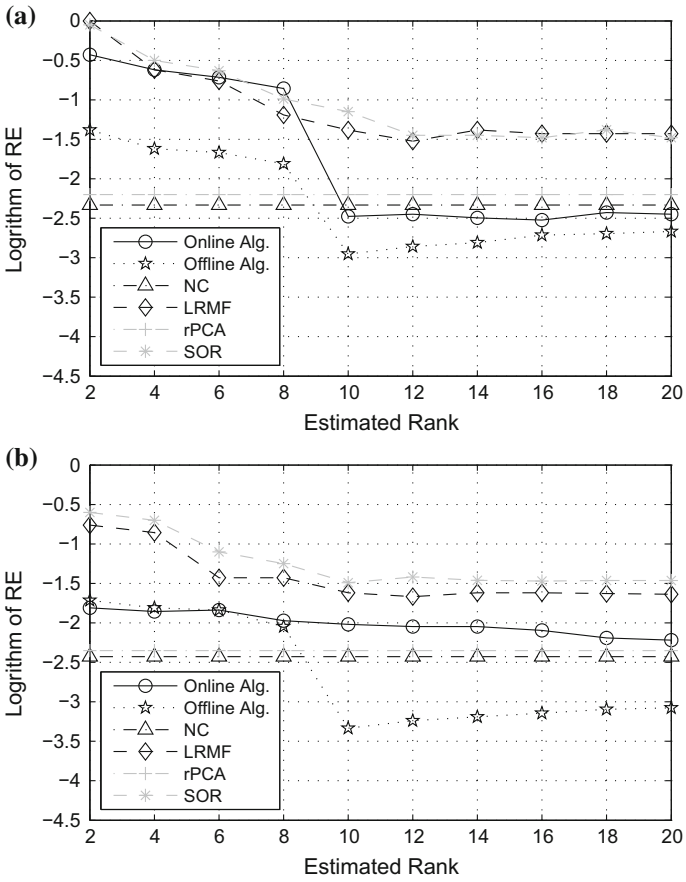

Fig. 7 Effect of upper bound r for the accuracy performance of our algorithms. **a** IEEE 300-bus system, **b** Polish power system

the accuracy performance of our algorithms. From the figure, we can observe that when the upper bound r equals true rank of the measurement matrix, our online and offline algorithms have the best performance; when r deviates from the true rank to a larger value, the accuracy performance of our algorithms decreases gracefully; when r decreases from the true value, the accuracy performance of our algorithms degenerates rapidly. Therefore, *in practice, we can relax the upper bound r to a relatively larger value to be safe*. As the NC and rPCA algorithms try to minimize the rank of real measurement matrix, they are not influenced by r.

In the large test case of Polish power system illustrated in Fig. 7b, the true rank of real measurement matrix is 10; similar results to those in the IEEE 300-bus system can be observed.

6.4 Efficiency study

We compare our algorithms with the NC, LRMF, rPCA, and SOR algorithms on computational complexity. We vary the rank of the measurement matrix on the Polish 3375-bus power system and test the CPU time required by each algorithm for convergence. The corrupted data ratio is fixed at $\epsilon = 0.1$.

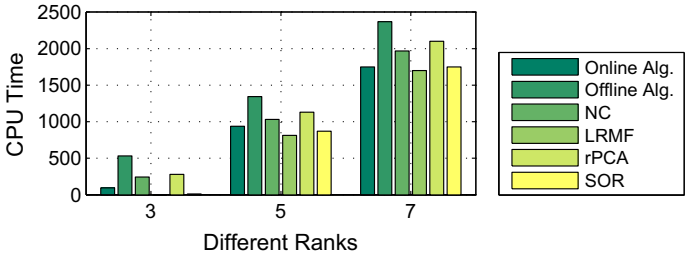

Fig. 8 CPU time comparison on Polish power system

As shown in Fig. 8, the total CPU time of our online algorithm is relatively lower than our offline algorithm. This can be explained by the fact that our online algorithm removes full SVD computation. The offline algorithm takes more time to arrive at its stationary point, as it combines SVD and ADMM. LRMF algorithm shows the best efficiency performance in this experiment. Nevertheless, our offline algorithm only takes about 20% more time than LRMF, while converging to a far more accurate result (see Fig. 7). Moreover, the increase in our online algorithm complexity is more moderate as the rank of real measurement matrix increases.

7 Conclusion

In this paper, we have modeled the problem of false data separation in a smart grid as a rank-constrained matrix optimization problem, which helps recover original real measurement data more accurately by preserving the true changes between the measurements at different time slots caused by the perturbations on grid states. We have also proposed both offline and online solutions for the problem. To the best of our knowledge, it is the first time that we explicitly solve the problem of online false data separation in a smart grid. Trace-driven simulations show that our approach compares favorably with the existing state-of-the-art solutions.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (61502347, 61502504, and 61628209), the Nature Science Foundation of Hubei Province of China (2016CFB384 and 2016CFA030), the Fundamental Research Funds for the Central Universities (2042015kf0038), the Research Funds of Renmin University of China (15XNLF09), and the Research Funds for Introduced Talents of Wuhan University.

References

1. Abur A, Exposito A G (2004) Power system state estimation: theory and implementation. Marcel Dekker Inc., New York
2. Bi S, Zhang YJ (2014) Graphical methods for defense against false-data injection attacks on power system state estimation. IEEE Trans Smart Grid 5(3):1216–1227
3. Bobba RB, Rogers KM, Wang Q, Khurana H, Nahrstedt K, Overbye, TJ(2010) Detecting false data injection attacks on dc state estimation. In: Preprints of the First Workshop on Secure Control Systems (SCS'10), CPSWEEK'10
4. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122
5. Candès EJ, Li X, Ma Y, Wright J (2011) Robust principal component analysis? J ACM 58(3):11

6. Caro E, Conejo AJ, Minguez R (2009) Power system state estimation considering measurement dependencies. IEEE Trans Power Syst 24(4):1875–1885
7. Ganeshy A, Wright J, Li X, Candes EJ, Ma Y (2010) Dense error correction for low-rank matrices via principal component pursuit. In: Proceedings of the 2010 IEEE International Symposium on Information Theory (ISIT'10), pp. 1513–1517
8. Haldar JP, Hernando D (2009) Rank-constrained solutions to linear matrix equations using powerfactorization. IEEE Signal Process Lett 16(7):584–587
9. Kim TT, Poor HV (2011) Strategic protection against data injection attacks on power grids. IEEE Trans Smart Grid 2(2):326–333
10. Kosut O, Jia L, Thomas RJ, Tong L (2010) On malicious data attacks on power system state estimation. In: Proceedings of the 45th International Universities Power Engineering Conference (UPEC'10), pp. 1–6
11. Kushner HJ, Yin GG (2003) Stochastic approximation and recursive algorithms and applications, vol 35. Springer, Berlin
12. Liu Y, Ning P, Reiter MK (2010) False data injection attacks against state estimation in electric power grids. ACM Trans Inf Syst Secur 14(1):21–32
13. Liu L, Esmalifalak M, Ding Q, Emesih VE, Han Z (2014) Detecting false data injection attacks on power grid by sparse optimization. IEEE Trans Smart Grid 5(2):612–621
14. Morales JM, Baringo L, Conejo AJ, Mínguez R (2010) Probabilistic power flow with correlated wind sources. IET Gener Transm Distrib 4(5):641–651
15. Niesen U, Shah D, Wornell GW (2009) Adaptive alternating minimization algorithms. IEEE Trans Inf Theory 55(3):1423–1429
16. Shen Y, Wen Z, Zhang Y (2014) Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. Optim Methods Softw 29(2):239–263
17. Siobhan G (2009) Electricity grid in U.S. penetrated by spies. http://online.wsj.com/articles/SB123914-805204099085
18. Wen Z, Yin W, Zhang Y (2012) Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math Program Comput 4(4):333–361
19. Xu W, Wang M, Tang A (2011) On state estimation with bad data detection. In: Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'11), pp. 5989–5994
20. Yang Q, Yang J, Yu W, An D, Zhang N, Zhao W (2014) On false data-injection attacks against power system state estimation: modeling and countermeasures. IEEE Trans Parallel Distrib Syst 25(3):717–729
21. Yuan X, Yang J (2009) Sparse and low rank matrix decomposition via alternating direction method. Pac J Optim 9(1):1–11

Hao Huang received a PhD degree in Computer Science from Zhejiang University, China, in 2012. He is currently an Associate Professor at State Key Laboratory of Software Engineering, Wuhan University, China. His research interests include data management and analytics, data mining, and intelligent information systems. He is a member of the ACM and the CCF.

Qian Yan received the B.E. degree in Computer Science from Wuhan University, China, in 2016. He is currently working toward the MS degree in Computer Science at State Key Laboratory of Software Engineering, Wuhan University, China. His research interests include data management and analytics, data mining, and intelligent information systems.

Yao Zhao received the B.E. degree from the Department of Electronics and Telecommunication Engineering, Harbin Institute of Technology, China, in 2013, and the MS degree in Computer Science from the University of Calgary, Canada, in 2015. He is currently working for a workforce analytics software company based in Vancouver, Canada. His research interests are in computer networks and smart grids.

Wei Lu is currently an associate professor at Renmin University of China. He received his PhD degree in Computer Science from Renmin University of China in 2011. His research interest includes query processing in the context of spatiotemporal, cloud database systems, and applications.

Zhenguang Liu is currently a postdoctoral research fellow with the Department of Computer Science, National University of Singapore, Singapore. He received the PhD degree from Zhejiang University and the BS degree from Shandong University, both in Computer Science. His research interests include data mining and intelligent systems.

Zongpeng Li received his B.E. degree in Computer Science and Technology from Tsinghua University (Beijing) in 1999, his M.S. degree in Computer Science from University of Toronto in 2001, and his PhD degree in Electrical and Computer Engineering from University of Toronto in 2005. His research interests are in computer networks, network coding, cloud computing, and energy networks. He was named an Edward S. Rogers Sr. Scholar in 2004, won the Alberta Ingenuity New Faculty Award in 2007, and was nominated for the Alfred P. Sloan Research Fellow in 2007. He received the "Outstanding Young Computer Science Researcher" Prize from the Canadian Association of Computer Science.